

University of Groningen

Ontwerp van een computerpakket voor het oplossen van storingsontkoppelingsproblemen

Smedinga, Rein

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Early version, also known as pre-print

Publication date:
1982

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):
Smedinga, R. (1982). Ontwerp van een computerpakket voor het oplossen van storingsontkoppelingsproblemen.

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

ONTWERP VAN EEN

COMPUTERPAKKET

VOOR HET OPLOSSEN VAN

TORINGENTKOPPELINGEN

PROBLEMEN

NEIN. MEDINGA

ONTWERP VAN EEN GEBRUIKERSVRIENDELIJK
INTERACTIEF COMPUTERPAKKET VOOR HET
OPLOSSEN VAN MULTIVARIABELE ONTWERP-
PROBLEMEN OP HET GEBIED VAN DE
STORINGSONTKOPPELING.

R. Smedinga
afstudeerwerk 1982.

In de lineaire stationaire systeemtheorie bestaan legio problemen op het gebied van de storingsontkoppeling. De oplossing van zo'n probleem zowel als het nagaan of een oplossing überhaupt bestaat, beide zijn slechts mogelijk na berekening van een aantal zg. invariante deelruimten.

De hoofdzaak van dit afstudeerwerk is het implementeren van algoritmen op de computer, waarmee deze deelruimten kunnen worden berekend. Vervolgens kan dan van een aantal storingsontkoppelingsproblemen worden nagegaan of ze oplosbaar zijn.

Dank ben ik verschuldigd aan:

J.C. Willems, voor het feit dat hij mijn interesse en belangstelling heeft gewekt voor de systeemtheorie in het algemeen en natuurlijk voor de hulp als afstudeerdocent.
C.J. van der Laan (Rekencentrum RUG), die mij bijzonder enthousiast heeft gemaakt voor de programmeertaal ALGOL68,
H.C. Croon, die dat enthousiasme terugbracht tot normale maatstaven en
D. de Vries (Rekencentrum RUG), voor de hulp die hij gaf bij problemen met ALGOL68.

INHOUD

inhoud	2
I. inleiding	3
II. invariante deelruimten	5
II.A. $A(\text{mod } B)$ invariante deelruimte	7
II.B. regelbaarheidsdeelruimte	8
II.C. $A \text{ ker } C$ invariante deelruimte	10
II.D. niet-waarneembaarheidsdeelruimte	12
II.E. dualiteit	13
II.F. algoritmes	15
II.G. Numerieke beschouwing	18
III. implementatie van de algoritmes	19
III.A. implementatie van ISA	20
III.B. implementatie van ACSA	26
III.C. bepaling van een $F \in \mathbb{F}_p(\mathbb{R}^* \text{ker } H)$	30
III.D. bepaling van een $L \in \mathbb{L}_p(\mathbb{N}^* \text{im } G)$	33
III.E. bepaling van een $K \in \mathbb{K}(V, S)$	36
III.F. poolplaatsingsalgoritme	38
IV. storingsontkoppelingsproblemen	41
IV.A. DDP: Disturbance Decoupling Problem	42
IV.B. DDEP: Disturbance Decoupling Estimation Problem	44
IV.C. DDPM: Disturbance Decoupling Problem with Measurements	46
V. bijna-storingsontkoppeling	51
V.A. Definities	52
V.B. implementaties	55
VI. voorbeelden	57
VI.A. destillatieskolom	58
VI.B. koersregeling bij zeeschepen	63
VI.C. een papierfabriek	67
lijst van gebruikte symbolen	75
lijst van gebruikte afkortingen	76
referenties	77
appendices:	
A. programmeren in ALGOL 68	79
B. handleiding LIDIA	83

hoofdstuk I. INLEIDING

Ter ondersteuning van het onderwijs in de systeemtheorie is de laatste jaren een aantal computerprogramma's geschreven, die allen sterk interactief werken. Dit houdt in dat men als gebruiker geen kennis van programmeren behoeft te bezitten, doch rechtstreeks met het programma communiceert door bepaalde opdrachten in te typen en gegevens in te voeren.

De aanzet hiertoe is wel gegeven door het programmapakket LINPAK, het afstudeerwerk van Geert Kamphuis (oktober 1979, zie /1/). Dit is een pakket voor het ontwerpen en analyseren van lineaire stationaire systemen.

Het pakket heeft de volgende mogelijkheden:

- invoeren van een systeem in toestandsvorm middels matrices A, B, C en D.
- nagaan van regel- en waarneembaarheid, alsmede stabiliteit en bepalen van openlooppolen.
- bepalen van de transfermatrix.
- poolplaatsing, berekening van feedbackmatrices.
- simulatie van flows.

In ontwerp of inmiddels voltooid zijn uitbreidingen van dit pakket met systeemrepresentaties in andere dan toestandsvorm en LQG-algoritmen voor het oplossen van optimal-control-problemen.

Mijn afstudeerwerk houdt eveneens een uitbreiding van LINPAK in. Dez ligt op het gebied van de multivariabele theorie (o.a. Wonham, zie /2/) en bepaalt oplossingen van o.a. de storingsontkoppelingproblemen DDP, DDEP en DDPM (resp. disturbance decoupling problem, disturbance decoupling estimation problem en disturbance decoupling problem with measurements).

Omdat de uitbreiding bedoeld was te zijner tijd opgenomen te worden in het bestaande pakket LINPAK is een en ander geschreven in de programmeertaal ALGOL 68, waardoor bovendien het TORRIX-pakket gebruikt kan worden. Wegens beperkingen van de CYBER ten aanzien van het geheugengebruik is het nodig de segmentloader te gebruiken. Zie voor een gedetailleerde beschrijving van de problemen met de CYBER bijlage A.

Enmaal werkend aan het pakket bleek echter dat de uitbreidingen zo omvangrijk waren dat het inpassen in LINPAK onoverkomelijke problemen zou gaan geven m.b.t. het geheugengebruik, zodat is besloten de implementatie als zelfstandig werkend pakket te voltooien. Hiervoor zijn enkele procedures uit LINPAK overgenomen.

Het pakket, dat de naam LIDIA kreeg (LIDIA = Linear systems Disturbance decoupling Algorithms), is volledig compatibel met LINPAK, zodat onder LINPAK opgegeven data daarna ook onder LIDIA kan worden geanalyseerd. Omgekeerd geldt dit ook. De matrix D in de systeembeschrijving $\dot{x} = Ax + Bu$, $y = Cx + Du$ wordt dan gevuld met nullen. Deze matrix heeft in LIDIA verder geen betekenis.

Het hoofddoel van LIDIA is het berekenen van de volgende deelruimtes:

- V^* ker H, de grootste (A,B)-invariante deelruimte die bevat is in ker H.
- R^* ker H, de grootste regelbaarheidsdeelruimte die bevat is in ker H.
- S^* im G, de kleinste (A,C)-invariante deelruimte die im G omvat.
- N^* im G, de kleinste niet-waarneembaarheidsdeelruimte die im G omvat.

Uitgaande van de systeembeschrijving:

$$\dot{x} = Ax + Bu + Gd$$

$$y = Cx$$

$$z = Hx$$

waar behalve de toestand x, de imput u en de output y ook zijn toegevoegd storingen d en te regelen c.q. te schatten variabelen z, wordt mbv. die deelruimtes de oplosbaarheid nagegaan van de problemen DDP, DDEP en DDPM en hun equivalenten waarbij poolplaatsingseisen zijn toegevoegd.

Indien het probleem inderdaad oplosbaar is wordt bovendien een oplossing bepaald.

Het pakket LIDIA kan in de toekomst nog worden uitgebreid met de zg. 'almost' versies van DDP, DDEP en DDPM, nl. ADDP, ADDEP en ADDPM (zie /12/ en /13/). In LIDIA kunnen al wel de 'almost' versies van de hierboven vermelde deelruimtes worden bepaald.

hoofdstuk II. INVARIANTE DEELRUIMTEN

In de systeemtheorie is het gebruikelijk, lineaire stationaire systemen voor te stellen d.m.v. de volgende zg. toestandsvergelijkingen:

$$\begin{aligned}\dot{x} &= A x + B u \\ y &= C x + D u\end{aligned}\quad (1)$$

waarbij \dot{x} staat voor $\frac{dx}{dt}$.

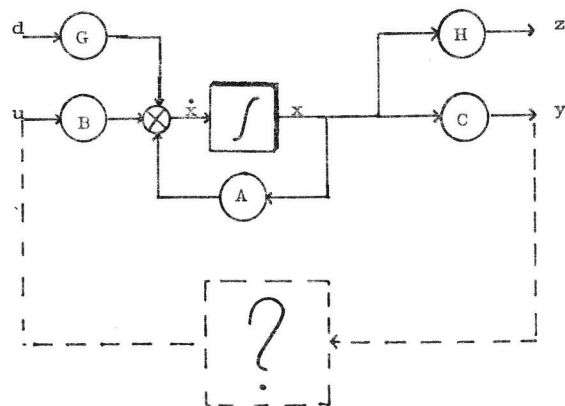
Bij storingsontkoppelingproblemen is het gebruikelijk de zg. 'direct link' D weg te laten. Het zo gewijzigde systeem (1) wordt vervolgens met storing uitgebreid en er wordt nog een extra variabele toegevoegd. We krijgen dan:

$$\begin{aligned}\dot{x} &= A x + B u + G d \\ y &= C x \\ z &= H x\end{aligned}\quad (2)$$

Hierbij is:

$$\begin{aligned}x \in X &= \mathbb{R}^n \text{ de toestand} \\ y \in Y &= \mathbb{R}^p \text{ de waarneming} \\ u \in U &= \mathbb{R}^m \text{ de sturing} \\ d \in D &= \mathbb{R}^q \text{ de storing} \\ z \in Z &= \mathbb{R}^r \text{ de te regelen variabele}\end{aligned}\quad (3)$$

Een voorbeeld van een storingsontkoppelingprobleem is nu het vinden van een zg. feedbackprocessor, die uitgaande van de waarneming y een sturing u bepaald, zodanig dat de te regelen variabelen z onafhankelijk zijn van de storingen d . (Dit is DDPM: Disturbance Decoupling Problem with Measurement)



Behalve DDPM bestaat er ook een vereenvoudigde versie (DBP) waarbij de gehele toestand wordt waargenomen ($C = I$).

Verder bestaan er nog legio problemen, die allen gebaseerd zijn op het idee de storing te ontkoppelen. Voorbeelden zijn te vinden in [2].

De oplossingen van al deze storingsontkoppelingproblemen gaan via het bepalen van een aantal deelruimten en het aan de hand van deze deelruimten nagaan van oplosbaarheid en bepalen van een eventuele oplossing.

In de volgende paragrafen zullen de vier benodigde deelruimten worden gedefinieerd.

notatie

Matrices zullen steeds worden aangegeven met hoofdletters (A), vectoren met kleine letters (a), lineaire deelruimten met onderstreepte hoofdletters (\underline{A}). Verder geldt:

$$\begin{aligned}\langle \underline{A} | \underline{B} \rangle &= \sum_{i=1}^{n-1} A^i \underline{B} \\ \langle \underline{K} | \underline{A} \rangle &= \bigcap_{i=1}^{n-1} A^{-i} \underline{K}\end{aligned}$$

Vaak zullen we \underline{B} gebruiken voor $\text{im } B$ en \underline{K} voor $\text{Ker } C$, terwijl $A|_{\underline{L}}$ de (afbeeldings)matrix A voorstelt beperkt tot de ruimte \underline{L} en $A(\text{mod } \underline{L})$ deze matrix A modulo \underline{L} .

A^* staat voor geadjungeerde matrix (welke ingeval we te maken hebben met reële matrices kunnen vervangen door A^t , de getransponeerde matrix).

\underline{L}^\perp staat voor het orthogonale complement van de deelruimte \underline{L} .

Daar waar sprake is van poolplaatsing verstaan we onder de polen van een systeem als in (1) de eigenwaarden van de matrix A (het spectrum van A), welke wordt aangeduid middels $\text{spec}(A)$. Zoals gebruikelijk is in de systeemtheorie wordt van dit spectrum geëist dat het symmetrisch is, d.w.z. dat als $z \in \text{spec}(A)$ ook geldt dat $\bar{z} \in \text{spec}(A)$.

Onder een trajectory $x(t)$ van een systeem als (1) verstaan we een functie $x(t)$ zodanig dat voor zekere $u(t)$ voldaan is aan $\dot{x}(t) = A x(t) + B u(t) \quad \forall t$.

Onder het karakteristieke polynoom van een matrix A verstaan we het polynoom $|A - \lambda I|$ en wordt aangegeven door χ_A .

Tenslotte willen we opmerken dat overal waar in de rest van de tekst gesproken wordt van transformatie of transformatiematrix steeds bedoeld wordt een orthogonale transformatie(matrix).

II.A A(mod B)-invariante deelruimte

Beschouw het systeem: $\dot{x} = A x + B u$.

Een deelruimte $\underline{V} \subset \underline{X}$ heet dan een A(mod B)- of (A,B) invariante deelruimte als:

- i. (gestuurde invariante of invarianteerbare deelruimte)
Voor $\forall x_0 \in \underline{V} \exists$ trajectory $x(t)$ zodanig dat $x(0) = x_0$ en $x(t) \in \underline{V} \forall t$.
- ii. (controlled invariant subspace)
 $A \cdot \underline{V} \subset \underline{V} + \underline{B}$
- iii. (feedback invarianteerbare deelruimte)
 $\exists F$ zodanig dat $(A + B \cdot F) \cdot \underline{V} \subset \underline{V}$

bewijs van equivalentie

- i. \rightarrow ii. : Kies een $x_0 \in \underline{V}$. Uit i. volgt dat

$$\dot{x}_0 = A x_0 + B u(0) \in \underline{V} \quad (1)$$
 (volgt uit $z \in \underline{Z}$ dan ook $\dot{z} \in \underline{Z}$)
 (1) impliceert:

$$A x_0 = \dot{x}_0 - B u(0) \in \underline{V} + \underline{B} \quad (2)$$
- ii. \rightarrow iii. : Kies een basis (v_1, \dots, v_k) voor \underline{V} . Uit ii. volgt:

$$A v_i = v_i' + b_i \quad \text{met } v_i' \in \underline{V} \text{ en } b_i \in \underline{B} \quad (3)$$

$$b_i \in \underline{B} \text{ duidt op het bestaan van een } u_i \in \underline{U} \text{ met } b_i = B \cdot u_i \quad (4)$$
 Definieer nu $F : \underline{V} \rightarrow \underline{U}$ door:

$$\left. \begin{aligned} F \cdot v_i &= u_i \quad i = 1, \dots, k \\ F \text{ willekeurig op } \underline{V}^\perp \end{aligned} \right\} \quad (5)$$
 Uit ii., samen met (3)-(5) volgt nu:

$$A \cdot v_i = v_i' + b_i = v_i' + B \cdot u_i = v_i' - B \cdot F \cdot v_i$$
 waaruit iii. volgt.
- iii. \rightarrow i. : triviaal, neem $u = F \cdot x$.

grootste element

Geef met \underline{V} de verzameling van al dergelijke \underline{V} 's aan. Noteer $\underline{V}(\underline{K}) = \{ \underline{V} \subset \underline{V} \mid \underline{V} \subset \underline{K} \}$. Dan is $\underline{V}(\underline{K})$ gesloten onder optelling:
 Laat $\underline{V}_1, \underline{V}_2 \in \underline{V}(\underline{K})$. Dan is $\underline{V}_1 + \underline{V}_2 \subset \underline{K}$ en

$$A \cdot (\underline{V}_1 + \underline{V}_2) = A \cdot \underline{V}_1 + A \cdot \underline{V}_2 = \underline{V}_1 + \underline{B} + \underline{V}_2 + \underline{B} = (\underline{V}_1 + \underline{V}_2) + \underline{B}$$
 zodat $(\underline{V}_1 + \underline{V}_2) \in \underline{V}(\underline{K})$.

Uit het gesloten zijn onder optelling volgt het bestaan van een grootste element, aangegeven met $\underline{V}^*(\underline{K})$. We hebben dus:

$$\underline{V}^*(\underline{K}) = \sup \{ \underline{V} \in \underline{V}(\underline{K}) \} \in \underline{V}(\underline{K}). \quad (6)$$

friend of \underline{V}

Voorts definiëren we:

$$\underline{F}(\underline{V}) = \{ F \mid (A + B \cdot F) \cdot \underline{V} \subset \underline{V} \} \quad (7)$$

Dergelijke matrices F worden, in analogie met /2/, 'friends' van \underline{V} genoemd.

II.B Regelbaarheidsdeelruimte

Beschouw wederom: $\dot{x} = A x + B u$

Een deelruimte $\underline{R} \subset \underline{X}$ heeft dan een regelbaarheidsdeelruimte, als:

- i. $\underline{R} \in \underline{V}$ en voor $\forall x_0, x_1 \in \underline{R} \exists T > 0$ en trajectory $x(t)$ zodanig dat $x(0) = x_0$, $x(T) = x_1$, en $x(t) \in \underline{R} \forall t$.
- ii. (bereikbare deelruimte na feedback met beperkte input)
 $\underline{R} \in \underline{V}$ en $\exists F$ en deelruimte $\underline{B}_1 \subset \text{im } B$ zodanig dat

$$\underline{R} = \langle A + B \cdot F \mid \underline{B}_1 \rangle$$
- iii. $\underline{R} \in \underline{V}$ en \forall polynomen p met graad $p = \dim \underline{R} \exists F$ zodanig dat:

$$\chi_{(A + B \cdot F) \mid \underline{R}} = p$$

bewijs van equivalentie:

- ii. \rightarrow i. De bereikbare verzameling (is de verzameling van alle toestanden x_1 , waarvoor een $T > 0$ en een trajectory $x(t)$ bestaat zodat $x(T) = x_1$) van het systeem

$$\dot{x} = (A + B \cdot F) x + B_1 u \quad (\text{met } \text{im } B_1 = \underline{B}_1) \quad (1)$$
 is juist gelijk aan \underline{R} (zie /2/, pagina 38).
 Aangezien ieder trajectory van (1) er ook een is van $\dot{x} = A x + B u$ (immers, er wordt slechts beperkte input toegelaten en feedback toegepast) is hiermee i. uit ii. bewezen.
- i. \rightarrow ii. Neem $\underline{B}_1 = \underline{B} \wedge \underline{R}$ en $F \in \underline{F}(\underline{R})$. We bewijzen dat dan:

$$\underline{R} = \langle A + B \cdot F \mid \underline{B}_1 \rangle \quad (2)$$
 Veronderstel $x \in \underline{R}$. Dan is $\dot{x} = A x + B u \in \underline{R}$
 Pas feedback $u = F x + u'$ toe met u' zodanig dat $B u' = B u$.
 Dan is nog steeds $\dot{x} = (A + B \cdot F) \cdot x + B_1 u' \in \underline{R}$
 waaruit volgt (omdat door de keuze van F ook $(A + B \cdot F) \cdot x \in \underline{R}$) dat $B_1 u' \in \underline{R}$.
 Per definitie is ook $B_1 u' \in \langle A + B \cdot F \mid \underline{B}_1 \rangle$ zodat $\underline{R} \subset \langle A + B \cdot F \mid \underline{B}_1 \rangle$.
 Omgekeerd geldt ook. Als namelijk $x \in \langle A + B \cdot F \mid \underline{B}_1 \rangle$ dan volgt (bijvoorbeeld met de bekende stelling van Cayley-Hamilton) dat $x = \exp(A + B \cdot F) \cdot B_1$ en dus een element van \underline{R} .
- ii. - iii. : zie /2/ pagina's 105 en 106.

grootste element

Geef met \underline{R} de verzameling van al dergelijke \underline{R} 's aan. Noteer $\underline{R}(\underline{K}) = \{ \underline{R} \in \underline{R} \mid \underline{R} \subset \underline{K} \}$. Dan is $\underline{R}(\underline{K})$ gesloten onder optelling.

Het gesloten zijn van \mathcal{R} volgt het snelste uit de eerste definitie samen met het volgende:

Laat $R_1, R_2 \in \mathcal{R}$. D.w.z. voor $x_0^i, x_1^i \in R_i$ bestaat $T_i > 0$ en trajectory $x^i(t)$, zodanig, dat $x^i(0) = x_0^i$ en $x^i(T_i) = x_1^i$ terwijl $x^i(t) \in R_i \forall t$. Veronderstel $T_1 < T_2$.

De trajectory $x(t)$, gedefinieerd door

$$x(t) = x^1(t) + x^2(t) \quad 0 \leq t \leq T_1$$

$$= x^1(T_1) + x^2(t) \quad T_1 \leq t \leq T_2$$

is dan zo, dat $x(t) \in R \forall t$ en bovendien is $x(0) = x_0^1 + x_0^2$ en $x(T_2) = x_1^1 + x_1^2$.

Uit het gesloten zijn volgt weer het bestaan van een grootste element. We vinden weer:

$$\underline{R}^*(K) = \sup \{ \underline{R} \in \mathcal{R}(K) \} \in \mathcal{R}(K) \quad (3)$$

friend of R

Uit de derde definitie volgt het bestaan van een matrix F , zodanig dat $\underline{R}^*(A + BF)$ -invariant is terwijl bovendien de polen in \underline{R}^* specificeerbaar zijn. D.w.z. bij vooraf gegeven polen is F zo te bepalen, dat $\text{spec}((A + BF)|\underline{R}^*)$ gelijk is aan die polen.

We definiëren daarom de volgende family matrices F :

$$\underline{F}_p(\underline{R}) = \{ F \mid (A + B \cdot F) \cdot \underline{R} \subset \underline{R} \text{ en } \text{spec}((A + B \cdot F)|\underline{R}) \text{ specificeerbaar} \} \quad (4)$$

II.C $A|_{\ker C}$ -invariante deelruimten

Beschouw het systeem $\dot{x} = A x, y = C x$.

Een deelruimte $\underline{S} \subset X$ heet dan een $A|_{\ker C}$ - of (A, C) -invariante deelruimte als:

i. (conditional invariant subspace)

\exists matrices F en E , zodanig dat $z := x \pmod{\underline{S}}$ voldoet aan

$$\dot{z} = F \cdot z + E \cdot y$$

ii. $A \cdot (\underline{S} \cap \ker C) \subset \underline{S}$

iii. (outputterugkoppelings-invariante deelruimte)

$\exists L$ zodanig dat $(A + L \cdot C) \underline{S} \subset \underline{S}$

bewijs van equivalentie

Hierbij gebruiken we de volgende relatie tussen F, E, L en A en C :

$$F = (A + L \cdot C) \pmod{\underline{S}} \quad (1)$$

$$E = -L \pmod{\underline{S}}$$

i. \rightarrow ii. : Stel $x \in \ker C$ (dus $y = C \cdot x = 0$). Uit i. volgt dan:

$$\dot{x} \pmod{\underline{S}} = \frac{d}{dt} x \pmod{\underline{S}} = \dot{z} = F \cdot z + E \cdot y = F \cdot z = F \cdot x \pmod{\underline{S}} = A \cdot x \pmod{\underline{S}}$$

waaruit $A x \pmod{\underline{S}} = \dot{x} \pmod{\underline{S}}$ en dus ii. volgt.

ii. \rightarrow iii. : Kies basis (s_1, \dots, s_k) voor \underline{S} met

$$s_1, \dots, s_q \in \underline{S} \setminus (\underline{S} \cap \ker C) \text{ en} \quad (2)$$

$$s_{q+1}, \dots, s_k \in \underline{S} \cap \ker C$$

Dan is $C \cdot s_i = y_i \quad i = 1, \dots, q$ voor zekere $y_i \in Y$

Definieer nu $L : Y \rightarrow X$ door

$$\left. \begin{aligned} L \cdot y_i &= -A \cdot s_i \quad i = 1, \dots, q \\ L &\text{ willekeurig op rest van } X \end{aligned} \right\} \quad (3)$$

Uit ii., samen met (2) en (3) volgt nu:

$$(A + L \cdot C) \cdot \underline{S} = (A + L \cdot C) \cdot (s_1, \dots, s_q, s_{q+1}, \dots, s_k) =$$

$$(A + L \cdot C) \cdot (s_1, \dots, s_q) + A \cdot (s_{q+1}, \dots, s_k) = 0 + \underline{S}$$

iii. \rightarrow i : Direct duidelijk uit:

$$\frac{d}{dt} x \pmod{\underline{S}} = \dot{x} \pmod{\underline{S}} = (A \cdot x) \pmod{\underline{S}} =$$

$$((A + L \cdot C) \cdot x) \pmod{\underline{S}} - L \pmod{\underline{S}} \cdot y =$$

$$(A + L \cdot C) \pmod{\underline{S}} \cdot x \pmod{\underline{S}} - L \pmod{\underline{S}} \cdot y =$$

$$F \cdot z + E \cdot y$$

kleinste element

Geef met \mathcal{S} de verzameling van al dergelijke \underline{S} 's aan. Noteer $\mathcal{S}(K) = \{ \underline{S} \in \mathcal{S} \mid K \subset \underline{S} \}$. Dan is $\mathcal{S}(K)$ gesloten onder intersectie. Laat $\underline{S}_1, \underline{S}_2 \in \mathcal{S}(K)$. Dan is $A \cdot ((\underline{S}_1 \cap \underline{S}_2) \cap \ker C) = A \cdot ((\underline{S}_1 \cap \ker C) \cap (\underline{S}_2 \cap \ker C)) \subset \underline{S}_1 \cap \underline{S}_2$, terwijl ook $K \subset \underline{S}_1 \cap \underline{S}_2$.

Uit het gesloten zijn onder intersectie volgt het bestaan van een kleinste element, aangegeven met $\underline{S}^*(K)$. We hebben nu:

$$\underline{S}^*(\underline{K}) = \inf \left\{ \underline{S} \in \mathcal{S}(\underline{K}) \right\} \in \mathcal{S}(\underline{K}) \quad (4)$$

friend of S

Ook nu definiëren we:

$$\underline{L}(\underline{S}) = \left\{ L \mid (A + L \cdot C) \cdot \underline{S} \subset \underline{S} \right\} \quad (5)$$

Dergelijke matrices L worden 'friends' van \underline{S} genoemd. De benaming 'friend of' wordt hier gebruikt in navolging van Wonham in /2/.

II.D Niet-waarneembaarheidsdeelruimte

Beschouw wederom: $\dot{x} = A x, y = C x$

Een deelruimte $\underline{N} \subset \underline{X}$ heet dan een niet-waarneembaarheidsdeelruimte als:

i. $\underline{N} \in \mathcal{S}$ en \exists matrix L en deelruimte $\underline{K}_1 \supset \ker C$, zodanig dat

$$\underline{N} = \langle \underline{K}_1 \mid A + L \cdot C \rangle$$

ii. $\underline{N} \in \mathcal{S}$ en \forall polynomen p met graad = toestandsdimensie -

$$\dim \underline{N} \exists \text{ matrix } L \text{ zodanig dat } \chi_{(A+LC)(\text{mod } \underline{N})} = p.$$

iii. $\underline{N} \in \mathcal{S}$ en het paar $((A+LC)(\text{mod } \underline{N}), C(\text{mod } \underline{N}))$ is waarneembaar voor alle matrices L , die aan i. of ii. voldoen.

bewijs van equivalentie

De equivalentie van bovenstaande definities volgt ogenblikkelijk uit de dualiteit van regelbaarheidsdeelruimte en niet-waarneembaarheidsdeelruimte. Zie II.E.

kleinste element

Geef met \mathcal{V} de verzameling van al dergelijke \underline{N} 's aan. Noteer $\mathcal{V}(\underline{K}) = \{ \underline{N} \in \mathcal{V} \mid \underline{K} \subset \underline{N} \}$. Dan is $\mathcal{V}(\underline{K})$ gesloten onder intersectie. (Volgt weer uit dualiteit met regelbaarheidsdeelruimte). Dit impliceert weer:

$$\underline{N}^*(\underline{K}) = \inf \left\{ \underline{N} \in \mathcal{V}(\underline{K}) \right\} \in \mathcal{V}(\underline{K}) \quad (1)$$

friend of N

Verder definiëren we weer:

$$\underline{L}_p(\underline{N}) = \left\{ L \mid (A + L \cdot C) \cdot \underline{N} \subset \underline{N} \text{ en spec } ((A+LC)(\text{mod } \underline{N})) \text{ specificieerbaar} \right\} \quad (2)$$

II.E Dualiteit

Onder dualiteit wordt verstaan, populair gezegd, de situatie die optreedt, als van een systeem de input en de output worden verwisseld. Voor ons is slechts van belang te onderkennen dat het systeem, beschreven door:

$$\begin{cases} \dot{x}_1 = A \cdot x_1 + B \cdot u_1 \\ y_1 = C \cdot x_1 \end{cases} \quad (1)$$

duaal is met het systeem:

$$\begin{cases} \dot{x}_2 = A^t \cdot x_2 + C^t \cdot u_2 \\ y_2 = B^t \cdot x_2 \end{cases} \quad (2)$$

In de lineaire algebra komt dualisatie neer op het omdraaien van pijlen in afbeeldingsdiagrammen en het transponeren van matrices.

Het is voor de rest van deze paragraaf voldoende, in te zien, eventueel per definitie, dat de systemen (1) en (2) elkaars duale zijn. Voor een uitgebreidere theorie, zie /6/ en /11/.

In de rest van deze paragraaf wordt gebruik gemaakt van het volgende lemma's:

lemma 1

Voor een deelruimte $\underline{W} \subset \underline{X}$ en een (matrix)afbeelding $M : \underline{X} \rightarrow \underline{X}$ geldt:

$$M \cdot \underline{W} \subset \underline{W} \quad \Leftrightarrow \quad M^t \cdot \underline{W}^\perp \subset \underline{W}^\perp$$

bewijs

Pas een basistransformatie toe zodat $T^{-1}x = (x_1, x_2)^t$ met $x_1 \in \underline{W}$ en $x_2 \in \underline{W}^\perp$.

Dan is:

$$T^{-1} \cdot M \cdot T = \begin{bmatrix} M_{11} & M_{12} \\ 0 & M_{22} \end{bmatrix} \quad \Leftrightarrow \quad T^{-1} \cdot M^t \cdot T = \begin{bmatrix} M_{11}^t & 0 \\ M_{12}^t & M_{22}^t \end{bmatrix}$$

en volgt het lemma uit:

$$T^{-1} \cdot M \cdot T \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \bar{x}_1 \\ 0 \end{bmatrix} \quad \Leftrightarrow \quad T^{-1} \cdot M^t \cdot T \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{x}_2 \end{bmatrix}$$

Hierbij is gebruikt gemaakt van het feit dat de matrix T orthogonaal is, zodat $T^{-1} = T^t$.

lemma 2

Met dezelfde \underline{W} en M als in lemma 1 geldt:

$$\text{spec}(M | \underline{W}) = \text{spec}(M^t | (\text{mod } \underline{W}^\perp))$$

bewijs

Is direct duidelijk door gebruik te maken van dezelfde transformatie als in het bewijs van lemma 1.

onderlinge relaties

We kunnen nu de volgende onderlinge relaties onderscheiden:

- i. \underline{W} is een $A(\text{mod } \underline{B})$ -invariante deelruimte $\langle = \rangle$
 \underline{W}^\perp is een $A^t | \ker B^t$ -invariante deelruimte
- ii. \underline{W} is een regelbaarheidsdeelruimte (m.b.t. A en B) $\langle = \rangle$
 \underline{W}^\perp is een niet-waarneembaarheidsdeelruimte (m.b.t. A^t en B^t)
- iii. $\underline{R}^*(\underline{K}) \subset \underline{V}^*(\underline{K})$
- iv. $\underline{N}^*(\underline{K}) \supset \underline{S}^*(\underline{K})$
- v. $\underline{R}^*(\underline{K}) = \langle A + B \cdot F | \underline{B} \wedge \underline{V}^*(\underline{K}) \rangle$ met $F \in \underline{F}(\underline{V}^*(\underline{K}))$ en $\underline{B} = \text{im } B$
- vi. $\underline{N}^*(\underline{U}) = \langle \underline{K} + \underline{S}^*(\underline{U}) | A + L \cdot C \rangle$ met $L \in \underline{L}(\underline{S}^*(\underline{U}))$ en $\underline{K} = \ker C$
- vii. $\underline{F}(\underline{V}^*(\underline{K})) \subset \underline{F}(\underline{R}^*(\underline{K}))$
- viii. $\underline{L}(\underline{N}^*(\underline{K})) \subset \underline{L}(\underline{S}^*(\underline{K}))$

bewijs en opmerkingen

i. volgt uit:

$$F \text{ zodanig dat } (A + B \cdot F) \cdot \underline{V} \subset \underline{V} \quad \langle = \rangle$$

$$F^t \text{ zodanig dat } (A^t + F^t \cdot B^t) \cdot \underline{V}^\perp \subset \underline{V}^\perp$$

de equivalentie zelf volgt uit lemma 1.

Populair gezegd houdt i. in dat de (A, B) -invariante deelruimten en de (A, C) -invariante deelruimten elkaars duale zijn.

ii. volgt uit:

$$F \text{ zodanig dat } (A + B \cdot F) \cdot \underline{R} \subset \underline{R} \quad \langle = \rangle$$

$$F^t \text{ zodanig dat } (A^t + F^t \cdot B^t) \cdot \underline{R}^\perp \subset \underline{R}^\perp$$

$$\text{samen met spec } ((A + B \cdot F) | \underline{R}) = \text{spec } (A^t + F^t \cdot B^t | (\text{mod } \underline{R}^\perp))$$

Populair gezegd houdt ii. in dat de regelbaarheidsdeelruimten en de niet-waarneembaarheidsdeelruimten elkaars duale zijn.

iii. triviaal.

iv. triviaal

v. zie /2/, pagina 109

vi. Volgt direct uit v. m.b.v. dualiteitsoverwegingen.

vii. volgt uit:

$$(A + B \cdot F) \cdot \langle A + B \cdot F | \underline{B} \wedge \underline{V}^*(\underline{K}) \rangle \subset \langle A + B \cdot F | \underline{B} \wedge \underline{V}^*(\underline{K}) \rangle$$

Deze inclusie zegt, dat wanneer voor zekere matrix F voldaan is aan : $(A + B \cdot F) \cdot \underline{V}^*(\underline{K}) \subset \underline{V}^*(\underline{K})$ dat dan ook voldaan is aan: $(A + B \cdot F) \cdot \underline{R}^*(\underline{K}) \subset \underline{R}^*(\underline{K})$.

viii. volgt uit vii. m.b.v. dualiteitsoverwegingen of direct uit vi.

Analoog aan vii. zegt deze inclusie dat wanneer voor zekere matrix L geldt: $(A + L \cdot C) \cdot \underline{S}^*(\underline{K}) \subset \underline{S}^*(\underline{K})$ automatisch voldaan is aan $(A + L \cdot C) \cdot \underline{N}^*(\underline{K}) \subset \underline{N}^*(\underline{K})$.

De dualiteitseigenschappen i. en ii. spelen een rol bij het bepalen van algoritmen voor berekening van de verschillende deelruimten, terwijl vii. en viii. dat doen bij de bepaling van een matrix uit $\underline{F}(\underline{R})$ of $\underline{L}(\underline{N})$.

II.F Algoritmen

In /2/ komen we algoritmen tegen voor het bepalen van $\underline{V}^*(\underline{K})$ en $\underline{R}^*(\underline{K})$. Deze worden respectievelijk aangeduid met ISA en ACSA, het Invariant Subspace Algorithm en het Almost Controllability Subspace Algorithm. ACSA wordt in /2/ aangeduid met CSA (Controllability Subspace Algorithm). De term ACSA is hier echter beter op zijn plaats omdat ACSA direct gebruikt kan worden ter bepaling van $\underline{R}_a^*(\underline{K})$, de Almost-versie van $\underline{R}^*(\underline{K})$, zie hoofdstuk VI.

De algoritmen hebben de volgende gedaante:

$$\begin{aligned} \text{(ISA)} \quad \underline{V}_K^{u+1} &= \underline{K} \cap A^{-1}(\underline{V}_K^u + \text{im } B) \quad , \quad \underline{V}_K^0 = \underline{X}. \\ \text{(ACSA)} \quad \underline{R}_K^{u+1} &= \underline{K} \cap (A \underline{R}_K^u + \text{im } B) \quad , \quad \underline{R}_K^0 = 0. \end{aligned}$$

bewijs van ISA

We bewijzen dat geldt:

$$\underline{K} \supset \underline{V}^1 \supset \underline{V}^2 \supset \dots \supset \underline{V}^u \supset \underline{V}^{u+1} \supset \dots \supset \underline{V}^*(\underline{K}) \quad (1)$$

$$\underline{V}^u = \underline{V}^{u+1} \text{ impliceert: } \underline{V}^u = \underline{V}^*(\underline{K}) \quad (2)$$

Uit (1) en (2) volgt dat \underline{V}^u convergeert, met limiet $\underline{V}^\infty = \underline{V}^*(\underline{K})$ waarbij de limiet bereikt wordt na ten hoogste $u = \dim \underline{V}^*(\underline{K})$ stappen. (d.w.z. $\underline{V}^\infty = \underline{V}^{\dim \underline{V}^*(\underline{K})} = \underline{V}^*(\underline{K})$).

Het bewijs van (1) en (2) gaat in stappen:

i. $\underline{V}^u \supset \underline{V}^{u+1}$

Volgt triviaal uit de constructie, bijvoorbeeld door inductie.

ii. $\underline{V}^u = \underline{V}^{u+1} \Rightarrow \underline{V}^u = \underline{V}^\infty$

Is triviaal.

iii. \underline{V}^∞ bestaat.

Volgt direct uit $\underline{V}^u \subset \underline{X}$ en (1).

iv. $\underline{V}^\infty \supset \underline{V}$ voor $\forall \underline{V} \in \underline{V}(\underline{K})$.

M.b.v. inductie geldt $\underline{V}^u \supset \underline{V}$ voor $\forall \underline{V} \in \underline{V}(\underline{K})$, immers:

a. triviaal voor $u = 0$

b. $\underline{V}^{u+1} = \underline{K} \cap A^{-1}(\underline{V}^u + \text{im } B) \supset \underline{K} \cap A^{-1}(\underline{V} + \text{im } B) \supset \underline{K} \cap \underline{V} = \underline{V}$, waarbij de laatste stap volgt uit

$$A \cdot \underline{V} \subset \underline{V} + \text{im } B \Rightarrow \underline{V} \subset A^{-1}(\underline{V} + \text{im } B)$$

v. $\underline{V}^\infty \in \underline{V}(\underline{K})$

Immers uit de constructie van ISA volgt $\underline{V}^\infty \subset \underline{K}$ en

$\underline{V}^\infty \subset A^{-1}(\underline{V}^\infty + \text{im } B)$ en uit deze laatste inclusie volgt weer $A \cdot \underline{V}^\infty \subset \underline{V}^\infty + \text{im } B$.

vi. $\dim(\underline{V}^{u+1}) - \dim(\underline{V}^u) \geq 1$, tenzij $\underline{V}^u = \underline{V}^\infty$, waaruit volgt dat de limiet na hoogstens $u = \dim \underline{V}^*(\underline{K})$ stappen wordt bereikt. $\underline{V}^*(\underline{K})$ vinden we dus door ISA toe te passen.

bewijs van ACSA

We bewijzen dat geldt (m.b.t. (ACSA)):

$$0 \subset \underline{R}^1 \subset \dots \subset \underline{R}^u \subset \underline{R}^{u+1} \subset \dots \subset \underline{R}^*(\underline{K}) \quad (3)$$

$$\underline{R}^u = \underline{R}^{u+1} \text{ impliceert: } \underline{R}^u = \underline{R}^*(\underline{K}) \quad (4)$$

Hierbij is:

$$\underline{R}^*(\underline{K}) = \inf \{ \underline{P} \in \mathcal{P}(\underline{K}) \}, \text{ met } \mathcal{P}(\underline{K}) = \{ \underline{P} \mid \underline{P} = \underline{K} \cap (A \underline{P} + \text{im } B) \} \quad (5)$$

Uit (3) en (4) volgt dat \underline{R}^u convergeert, met limiet $\underline{R}^\infty = \underline{R}^*(\underline{K})$ waarbij de limiet bereikt wordt na ten hoogste $u = \dim \underline{R}^*(\underline{K})$ stappen. (d.w.z. $\underline{R}^\infty = \underline{R}^{\dim \underline{R}^*(\underline{K})} = \underline{R}^*(\underline{K})$).

Het bewijs gaat weer in stappen:

i. $\underline{R}^u \subset \underline{R}^{u+1}$

is weer triviaal en bijvoorbeeld te bewijzen m.b.v. inductie:

a. $u = 0$ triviaal.

b. $\underline{R}^{u+1} = \underline{K} \cap (A \underline{R}^u + \text{im } B) \subset \underline{K} \cap (A \underline{R}^{u-1} + \text{im } B) = \underline{R}^u$

ii. $\underline{R}^u = \underline{R}^{u+1} \Rightarrow \underline{R}^u = \underline{R}^\infty$

triviaal uit (1)

iii. \underline{R}^∞ bestaat.

Volgt uit $\underline{R}^u \subset \underline{X}$ en i.

iv. $\underline{R}^\infty \subset \underline{P}$ voor $\forall \underline{P} \in \mathcal{P}(\underline{K})$

M.b.v. inductie geldt $\underline{R}^u \subset \underline{P}$ voor alle $\underline{P} \in \mathcal{P}(\underline{K})$:

a. $u = 0$ triviaal

b. $\underline{R}^{u+1} = \underline{K} \cap (A \underline{R}^u + \text{im } B) \subset \underline{K} \cap (A \underline{P} + \text{im } B) = \underline{P}$

v. $\underline{R}^\infty \in \mathcal{P}(\underline{K})$.

triviaal.

vi. $\dim(\underline{R}^u) - \dim(\underline{R}^{u+1}) \geq 1$ tenzij $\underline{R}^u = \underline{R}^\infty$, waaruit volgt dat de limiet na hoogstens $u = \dim \underline{R}^*(\underline{K})$ stappen wordt bereikt.

$\underline{R}^*(\underline{K})$ volgt nu uit de gelijkheid:

$$\underline{R}^*(\underline{K}) = \underline{R}^*(\underline{V}^*(\underline{K})) \quad (5)$$

(5) volgt uit:

a. $\underline{R}^*(\underline{K}) \in \mathcal{P}(\underline{V}^*(\underline{K}))$, immers:

$$\underline{V}^*(\underline{K}) \cap (A \cdot \underline{R}^*(\underline{K}) + \text{im } B) =$$

$$\underline{V}^*(\underline{K}) \cap (A \cdot \langle A + B F \mid \underline{B} \wedge \underline{V}^*(\underline{K}) \rangle + \text{im } B) =$$

$$\underline{V}^*(\underline{K}) \cap \langle A + B F \mid \underline{B} \wedge \underline{V}^*(\underline{K}) \rangle = \underline{R}^*(\underline{K})$$

(met $F \in \underline{F}(\underline{V}^*(\underline{K}))$ en $\underline{B} = \text{im } B$)

b. $\forall \underline{P} \in \mathcal{P}(\underline{V}^*(\underline{K}))$ is: $\underline{R}^*(\underline{K}) \subset \underline{P}$, immers:

$$\underline{P} = \underline{V}^*(\underline{K}) \cap (A \underline{P} + \text{im } B) \supset \underline{R}^*(\underline{K}) \cap (A \underline{P} + \text{im } B) \supset$$

$$\underline{R}^*(\underline{K}) \cap (A \underline{P}^*(\underline{V}^*(\underline{K})) + \text{im } B) \supset \underline{R}^*(\underline{K}).$$

$\underline{R}^*(\underline{K})$ vinden we dus door ACSA toe te passen met de ruimte \underline{K} in het algoritme vervangen door $\underline{V}^*(\underline{K})$.

conclusie

Gegeven de algoritmen:

$$\begin{aligned} \text{(ISA)} \quad \frac{u+1}{\underline{V}_K} &= \underline{K} \cap A^{-1} \left(\frac{u}{\underline{V}_K} + \text{im } B \right), \quad \frac{o}{\underline{V}_K} = \underline{X} & \text{limiet } \underline{V}_K^\infty \\ \text{(ACSA)} \quad \frac{u+1}{\underline{R}_K} &= \underline{K} \cap (A \frac{u}{\underline{R}_K} + \text{im } B), \quad \frac{o}{\underline{R}_K} = 0 & \text{limiet } \underline{R}_K^\infty \end{aligned}$$

Dan geldt:

$$\underline{V}^*(\underline{K}) = \underline{V}_K^\infty \quad (6)$$

$$\underline{R}^*(\underline{K}) = \underline{R}_{\underline{V}^*(\underline{K})}^\infty \quad (7)$$

In /4/ zijn analoge algoritmen vermeld voor de berekening van de ruimten $\underline{S}^*(\underline{K})$ en $\underline{N}^*(\underline{K})$. Om echter niet onnodig te hoeven programmeren, is het ook mogelijk deze beide ruimten te vinden door toepassing van de dualiteit met $\underline{V}^*(\underline{K})$ en $\underline{R}^*(\underline{K})$. Pas daartoe ISA en ACSA toe op het duale systeem:

$$\begin{aligned} \text{(ISA)}' \quad \frac{u+1}{\underline{V}_K} &= \underline{K} \cap (A^t)^{-1} \left(\frac{u}{\underline{V}_K} + \text{im } C^t \right), \quad \frac{o}{\underline{V}_K} = \underline{X} \\ \text{(ACSA)}' \quad \frac{u+1}{\underline{R}_K} &= \underline{K} \cap (A^t \cdot \frac{u}{\underline{R}_K} + \text{im } C^t), \quad \frac{o}{\underline{R}_K} = 0 \end{aligned}$$

Samen met de eigenschappen i. en ii. in II.E is het niet moeilijk in te zien dat nu geldt:

$$\underline{S}^*(\underline{K}) = \left(\frac{\underline{V}^*(\underline{K})}{\underline{K}^\perp} \right)^\perp \quad (8)$$

$$\underline{N}^*(\underline{K}) = \left(\frac{\underline{R}^*(\underline{K})}{(\underline{S}^*(\underline{K}))^\perp} \right)^\perp$$

Hierbij duidt het accent ' op het feit dat het hier oplossingen betreft van (ISA)' resp. (ACSA)'.

Het is dus voldoende de beide algoritmen ISA en ACSA te implementeren. Een en ander moet dan wel op een numeriek verantwoorde manier gebeuren.

II.G. Numerieke beschouwing.

In eerste instantie zijn er (op dit moment) drie mogelijkheden om ISA en ACSA te implementeren:

- rechtstreeks
- mbv. eigenwaarden (Laub en Moore, zie /15/).
- mbv. singuliere waarden (v. Dooren, zie /3/).

Zoals Wonham in /2/ al opmerkt zijn beide algoritmen in de huidige vorm numeriek zeer onstabiel, omdat voortdurend deelruimten moeten worden opgeteld of doorsnijdingen moeten worden gemaakt. Een kleine verstoring kan er dan al de oorzaak van zijn dat het resultaat een dimensie hoger of lager uitpakt.

Methodes die berusten op het berekenen van eigenwaarden zijn o.a. te vinden in /15/. Ook deze methodes zijn nog niet volmaakt omdat eigenwaarden niet altijd even nauwkeurig berekend kunnen worden. Bovendien veronderstellen Laub en Moore in /15/ slechts de situatie $R^* \text{ker } C = 0$.

De singuliere waarden van een matrix zijn eenvoudiger en nauwkeuriger te bepalen dan de eigenwaarden. Bovendien geeft het algoritme ter bepaling van deze waarden op eenvoudige manier orthonormale transformaties waarmee de rijen resp. de kolommen van een matrix kunnen worden 'samengedrukt', dwz. de onafhankelijke rijen, resp. kolommen kunnen bij elkaar geveegd worden.

Omdat de derde methode numeriek het nauwkeurigst is, is voor die methode gekozen.

Voor een uitgebreidere beschouwing over de numerieke nauwkeurigheid van de gebruikte methode, wordt verwezen naar /3/.

hoofdstuk III. IMPLEMENTATIE VAN DE ALGORITMEN.

Voor de implementatie van de algoritmen (ISA) en (ACSA) is gekozen voor de methode zoals beschreven in /3/. Deze methode is gebaseerd op het SVD-algoritme.

Singular Value Decomposition.

Gegeven een willekeurige matrix A ($m \times n$) bestaat er altijd een decompositie van de vorm:

$$A = U \cdot S \cdot V^* \quad (1)$$

waarbij U ($m \times m$) en V ($n \times n$) orthogonale matrices zijn,

$$\text{d.w.z.: } U \cdot U^* = U^* \cdot U = I_m$$

$$V \cdot V^* = V^* \cdot V = I_n$$

en

$$S = \begin{bmatrix} S_p & 0 \\ 0 & 0 \end{bmatrix}, \text{ met } S_p = \text{diag}(s_1, \dots, s_p) \quad (2)$$

$$s_1 \geq s_2 \geq \dots \geq s_p \geq 0$$

Uit een en ander volgt direct dat p de rang van de matrix A is.

SVD levert ons nu een orthogonale transformatie voor het vinden van p onafhankelijke kolommen van A en evenzo voor p onafhankelijke rijen van A . Immers, uit (1) en (2) volgt:

$$U^* \cdot A = \begin{bmatrix} A_r \\ 0 \end{bmatrix} \quad \text{en} \quad A \cdot V = \begin{bmatrix} A_c & 0 \end{bmatrix} \quad (3)$$

Hierbij is A_r een ($p \times n$) matrix met p onafhankelijke rijen en

A_c een ($m \times p$) matrix met p onafhankelijke kolommen.

De decomposities in (3) worden in /3/ resp. rijsamendrukking en kolomsamendrukking genoemd. De implementatie van (ISA) en (ACSA) is niets anders dan herhaaldelijke toepassing van deze transformaties.

Van het SVD-algoritme bestaat een implementatie in de NAG-bibliotheek, waarvan wordt geclaimd dat deze een nauwkeurigheid heeft, zodanig, dat met Q , D en P de berekende matrices U , S resp. V geldt:

$$Q \cdot D \cdot P^t = A + E \quad (4)$$

$$\text{en } \|E\|_2 \leq c \cdot \text{eps} \cdot s_1$$

Hierbij is c een functie van m en n en eps de machine-nauwkeurigheid.

We mogen concluderen, dat het numerieke algoritme voldoende stabiel is voor onze toepassingen.

Opmerking: In de implementaties van de algoritmen zal i.p.v. de kolomsamendrukking zoals in (3) er een gebruikt worden van de vorm: $A \cdot \hat{V} = \begin{bmatrix} 0 & \hat{A}_c \end{bmatrix}$, met:

$$\hat{V} = V \cdot I_a \quad \text{en} \quad \hat{A}_c = A_c \cdot I_a \quad (5)$$

$$\text{met } I_a = \begin{bmatrix} 0 & & 1 \\ & \ddots & \\ 1 & & 0 \end{bmatrix}.$$

Met behulp van de transformaties, zoals beschreven in (3) is het eenvoudig mogelijk, van A de ruimtes $\text{im } A$ en $\text{ker } A$ te bepalen:

$$\text{ker } A = \text{laatste } m - p \text{ kolommen van } V \quad (6)$$

$$\text{im } A = \text{eerste } p \text{ kolommen van } U$$

De kolommen spannen steeds een orthonormale basis op, omdat U en V orthogonale matrices zijn.

Gebruiken we de transformatie (5) dan vinden we:

$$\text{ker } A = \text{eerste } m - p \text{ kolommen van } \hat{V} \quad (7)$$

Immers, als we met $I_p = (e_1, \dots, e_p)$ de eerste p eenheidsvectoren aangeven, dan vinden we:

$$\text{im } A = U \cdot \text{im } U^* \cdot A = U \cdot \text{im } \begin{bmatrix} A_r \\ 0 \end{bmatrix} = U \cdot \text{sp } I_p = \text{sp}(u_1, \dots, u_p)$$

en:

$$\text{ker } A = \text{ker } A \cdot \hat{V} \cdot \hat{V}^* = \text{ker } \begin{bmatrix} 0 & A_c \end{bmatrix} \cdot \hat{V}^* = \text{sp } I_p \cdot \hat{V}^* = \text{sp}(\hat{v}_1, \dots, \hat{v}_p)$$

Verder kunnen we met de transformaties (3) (of (5)) eenvoudig nagaan of voor zekere matrices C en D geldt:

$$\text{im } C \subset \text{im } D \quad (8)$$

Immers, veronderstel:

$$U^* \cdot D = \begin{bmatrix} D_r \\ 0 \end{bmatrix} \begin{matrix} \uparrow p \\ \downarrow m-p \end{matrix} \quad (9)$$

en laat verder:

$$U^* \cdot C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \begin{matrix} \uparrow p \\ \downarrow m-p \end{matrix} \quad (10)$$

dan is aan (8) voldaan, d.e.s.d.a. $C_2 = 0$.

Dit laatste nu vorm de basis van de implementatie van (ISA).

In denotatie wordt gebruik gemaakt van de systeemmatrix P , welke, gegeven het systeem

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

gedefinieerd wordt als:

$$P = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \quad (11)$$

In de literatuur wordt i.p.v. C meestal $-C$ gebruikt, maar aangezien dat minteken hier verder irrelevant is en slechts tot verwarring aanleiding kan geven is de gedaante (11) gebruikt.

III.A. implementatie van ISA

Berekend wordt een orthogonale basistransformatie T_j en een getal n_j zodanig dat de eerste n_j kolommen van de matrix T_j de ruimte V_j opspannen. Bewezen wordt dat V_j voldoet aan het algoritme van Wonham:

$$\begin{aligned} V_1 &= K \\ V_j &= K \cap A^{-1}(B + V_{j-1}) \quad j=2,3,\dots \end{aligned} \quad (1)$$

met $K = \ker C$ en $B = \text{im } B$.

(Met voldoen bedoelen we hier: als V_j exact berekend zou worden zoals in het algoritme beschreven staat, dan voldoet V_j exact aan (1). Omdat V_j niet exact bekend is, doch slechts numeriek wordt benaderd kunnen we onmogelijk eisen dat exact aan (1) voldaan wordt. We mogen echter wel stellen, dat V_j een goede numerieke benadering zal zijn voor de exacte oplossing van (1)).

Als het algoritme stopt hebben we een orthogonale matrix T_{∞} , een getal n_{∞} en spannen de eerste n_{∞} kolommen van T_{∞} de ruimte V_{∞} op. V_{∞} voldoet aan (1) en is daarmee een goede numerieke benadering voor $V^{\infty} \ker(C)$.

outline van het algoritme

a. gegeven: A ($n \times n$)
 B ($n \times m$) ofwel: $P = \begin{bmatrix} A & B \\ C & 0 \end{bmatrix}$
 C ($p \times n$)

b. bepaal T_1 ($n \times n$) en S_1 ($p \times cr$) zodanig dat:
 $C T_1 = \begin{bmatrix} 0 & S_1 \end{bmatrix}$

waarbij cr de kolomrang is van C .

c. herbeneemen:

$$\begin{bmatrix} T_1 & 0 \\ 0 & I_p \end{bmatrix}^{-1} P \begin{bmatrix} T_1 & 0 \\ 0 & I_m \end{bmatrix} = \begin{bmatrix} T_1^{-1} A T_1 & T_1^{-1} B \\ -C T_1^{-1} & 0 \end{bmatrix} = \begin{bmatrix} A_1 & * & B_1 \\ C_1 & * & D_1 \\ 0 & -S_1 & 0 \end{bmatrix} \begin{matrix} \updownarrow n_1 \\ \updownarrow p_1 \\ \updownarrow p \end{matrix}$$

$\xrightarrow{n_1} \quad \xrightarrow{cr} \quad \xrightarrow{m}$

$$\begin{aligned} n_1 &= n - cr \\ p_1 &= cr \\ d_1 &= 0 \\ j &= 2 \end{aligned}$$

d. bepaal U_j^* ($p_{j-1} \times p_{j-1}$) en D_{j-1}^1 ($rr \times m$) zodanig dat:

$$U_j^* D_{j-1}^1 = \begin{bmatrix} D_{j-1}^1 \\ 0 \end{bmatrix}$$

waarbij rr de rijrang is van D_{j-1} .

e. stop indien $p_{j-1} = rr$ (D_{j-1} heeft volle rijrang).

f. pas C_{j-1} aan:

$$U_j^* C_{j-1} = \begin{bmatrix} C_{j-1}^1 \\ C_{j-1}^2 \end{bmatrix}$$

g. bepaal V_j ($n_{j-1} \times n_{j-1}$) en S_j ($p_{j-1} - rr \times cr$) zodanig dat

$$C_{j-1}^2 V_j = \begin{bmatrix} 0 & S_j \end{bmatrix}$$

waarbij cr de kolomrang is van C_{j-1}^2

h. stop indien $cr = n_{j-1}$ (resultaat is de nulruimte)

stop indien $cr = 0$ (resultaat is gelijk aan dat van de vorige stap, bovendien is nu: $\text{im } C_{j-1} \subset \text{im } D_{j-1}$)

i. herbeneemen:

$$\begin{bmatrix} V_j & 0 \\ 0 & U_j^* \end{bmatrix} \begin{bmatrix} A_{j-1} & B_{j-1} \\ C_{j-1} & D_{j-1} \end{bmatrix} \begin{bmatrix} V_j & 0 \\ 0 & I_m \end{bmatrix} = \begin{bmatrix} V_j^* A_{j-1} & V_j^* B_{j-1} \\ U_j^* C_{j-1} & U_j^* D_{j-1} \end{bmatrix} \begin{bmatrix} V_j & 0 \\ 0 & I_m \end{bmatrix} =$$

$$\begin{bmatrix} V_j^* A_{j-1} & V_j^* B_{j-1} \\ C_{j-1}^1 & D_{j-1}^1 \\ C_{j-1}^2 & 0 \end{bmatrix} \begin{bmatrix} V_j & 0 \\ 0 & I_m \end{bmatrix} = \begin{bmatrix} V_j^* A_{j-1} V_j & V_j^* B_{j-1} \\ C_{j-1}^1 V_j & D_{j-1}^1 \\ C_{j-1}^2 V_j & 0 \end{bmatrix} =$$

$$\begin{bmatrix} V_j^* A_{j-1} V_j & V_j^* B_{j-1} \\ C_{j-1}^1 V_j & D_{j-1}^1 \\ 0 & S_j & 0 \end{bmatrix} \begin{matrix} \updownarrow n_{j-1} \\ \updownarrow rr \\ \updownarrow cr \end{matrix} = \begin{bmatrix} A_j & * & B_j \\ C_j & * & D_j \\ 0 & S_j & 0 \end{bmatrix} \begin{matrix} \updownarrow n_j \\ \updownarrow p_j \\ \updownarrow cr \end{matrix}$$

$\xrightarrow{n_j} \quad \xrightarrow{cr} \quad \xrightarrow{m}$

$$\begin{aligned} n_j &= n_{j-1} - cr \\ p_j &= cr + rr \\ d_j &= d_{j-1} + p_{j-1} - rr \end{aligned}$$

$$T_j = T_{j-1} \cdot \text{diag}(V_j, U_j, I_{d_{j-1}})$$

j. herhaal vanaf stap d.

Na iedere deelstap j geldt het volgende:

$$\begin{bmatrix} T_j & 0 \\ 0 & I_p \end{bmatrix}^{-1} P \begin{bmatrix} T_j & 0 \\ 0 & I_m \end{bmatrix} =$$

$$\begin{bmatrix} A_j & * & * & . & * & * & B_j \\ C_j & * & * & . & * & * & D_j \\ 0 & S_j & * & . & * & * & 0 \\ . & . & . & . & . & . & . \\ 0 & . & 0 & S_j & * & * & 0 \\ 0 & . & 0 & 0 & S_2 & * & 0 \\ 0 & . & . & 0 & 0 & S_1 & 0 \end{bmatrix} \begin{matrix} \updownarrow n_j \\ \updownarrow p_j \\ \updownarrow d_j \\ \updownarrow p \end{matrix}$$

$\xrightarrow{n_j} \quad \xrightarrow{n-n_j} \quad \xrightarrow{m}$

bewijs van het algoritme.

We schrijven $\text{im } I_{n_j} = \text{sp} (e_1, \dots, e_{n_j})$ voor het opspansel van de eerste n_j standaard eenheidsvectoren en analoog $S_j = T_j \cdot \text{im } I_{n_j} = \text{sp} (t_1, \dots, t_{n_j})$ voor het opspansel van de eerste n_j kolommen van T_j .

We bewijzen nu, dat deze S_j aan het algoritme van Wonham voldoet:

a. $S_1 = \ker C$

Dit is direct duidelijk uit de stappen a, b en c.

b. $S_j = \ker C \cap A^{-1}(\text{im } B + S_{j-1})$

Deze gelijkheid is equivalent met (linksvermenigvuldigen met resp. A en T_j^{-1}):

$$T_j^{-1} A \cdot T_j \cdot \text{im } I_{n_j} = T_j^{-1} A \cdot T_j \cdot T_j^{-1} \cdot T_1 \cdot \text{im } I_{n_1} \cap (\text{im } T_j^{-1} B + T_j^{-1} T_{j-1} \cdot \text{im } I_{n_{j-1}})$$

uitwerken hiervan geeft:

$$\begin{bmatrix} A_j & * & * & . & * \\ C_j & * & * & . & * \\ 0 & S_j & * & . & * \\ . & . & . & . & . \\ 0 & 0 & . & S_2 & * \end{bmatrix} \begin{bmatrix} * \\ * \\ 0 \\ . \\ 0 \end{bmatrix} = \begin{bmatrix} A_j & * & * & . & * \\ C_j & * & * & . & * \\ 0 & S_j & * & . & * \\ . & . & . & . & . \\ 0 & 0 & . & S_2 & * \end{bmatrix} \begin{bmatrix} * \\ . \\ . \\ * \\ 0 \end{bmatrix} \cap \begin{bmatrix} * \\ * \\ 0 \\ . \\ 0 \end{bmatrix} \quad (2)$$

$y_1 \qquad y_2 \qquad y_3$

waarbij de kolommen y_1 een symbolische voorstelling zijn:

$y_1 = \text{sp} (e_1, \dots, e_{n_j})$, $y_2 = \text{sp} (e_1, \dots, e_{n_1})$ en

$y_3 = \text{sp} (e_1, \dots, e_{n_j}, e_{n_j+1}, \dots, e_{n_j+p_j})$. Deze laatste volgt uit:

$$T_j \cdot B + T_j^{-1} T_{j-1} \cdot \begin{bmatrix} * \\ * \\ 0 \\ 0 \end{bmatrix} \uparrow_{n_{j-1}} = \begin{bmatrix} V_j^* \\ U_j^* \\ I \end{bmatrix} \begin{bmatrix} B_{j-1} \\ D_{j-1} \\ 0 \end{bmatrix} + \begin{bmatrix} * \\ 0 \\ 0 \\ 0 \end{bmatrix} \uparrow_{n_{j-1}}$$

$$\begin{bmatrix} n_{j-1} \uparrow \\ p_{j-1} \uparrow \end{bmatrix} = \begin{bmatrix} V_j^* B_{j-1} \\ U_j^* D_{j-1} \\ - \\ 0 \end{bmatrix} + \begin{bmatrix} * \\ 0 \\ - \\ 0 \end{bmatrix} = \begin{bmatrix} V_j^* B_{j-1} \\ D_{j-1} \\ - \\ 0 \end{bmatrix} + \begin{bmatrix} * \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} * \\ * \\ 0 \\ 0 \end{bmatrix} \uparrow_{n_j} = y_3$$

Beschouwen we nu de uitwerking (2) nader, dan zien we dat de eerst en tweede rij triviaal zijn, evenals de vierde en volgende rijen. Het enig essentiële zit in de derde rij:

$$\begin{bmatrix} 0 & S_j & * & . & * \end{bmatrix} \begin{bmatrix} * \\ 0 \\ . \\ . \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & S_j & * & . & * \end{bmatrix} \begin{bmatrix} * \\ * \\ . \\ * \\ 0 \end{bmatrix} \cap (0 + 0) = 0$$

Een n_j , die echt kleiner is dan de vorige n_{j-1} , kan dus

slechts gevonden worden, als de matrix D_{j-1} nog geen volle rijrang heeft en dus samengedrukt kan worden (waarbij D_{j-1}^1 volle rijrang heeft), terwijl voor C_{j-1} , die hierdoor overgaat in $\begin{bmatrix} C_{j-1}^1 \\ C_{j-1}^2 \end{bmatrix}$, geldt, dat C_{j-1}^2 nog echt samengedrukt kan

worden tot $[0 \ S_j]$. Verder stopt het algoritme ook, als $n_j = 0$

daar verder uitwerken dan geen zin heeft.

Met bovenstaande is bewezen, dat de gevonden deelruimte S_j voldoet aan:

$S_1 = \ker C$

$S_j = \ker C \cap A^{-1}(\text{im } B + S_{j-1})$

De eerste stap wijkt af van de eerste stap in het algoritme zoals Wonham dit geeft. Het is echter direct duidelijk, dat uit $S_0 = X$ volgt: $S_1 = \ker C \cap A^{-1}(\text{im } B + X) = \ker C$.

berekening van een $F \in F(V^* \ker C)$

Een matrix F , zodanig dat $(A + BF) \underline{V} \subset \underline{V}$ kan na beëindiging van ISA zeer snel worden gevonden:

Bepaal eerst de matrix F_0 ($m \times n_j$) zodanig dat:

$$D_j \cdot F_0 = -C_j \quad (1)$$

Dan is $F = \begin{bmatrix} F_0 & * \end{bmatrix} \cdot T^{-1}$

Hierbij is $*$ willekeurig.

bewijs:

Met $\underline{V} = T \cdot \text{im } I_{n_j}$, ingevuld in $(A + BF) \cdot \underline{V} \subset \underline{V}$ vinden we:

$$(A + B \cdot F) \cdot T \cdot \text{im } I_{n_j} \subset T \cdot \text{im } I_{n_j}$$

Links vermenigvuldigen met T^{-1} geeft:

$$(T^{-1} A \cdot T + T^{-1} B \cdot F \cdot T) \text{im } I_{n_j} \subset \text{im } I_{n_j}$$

Met $F = \begin{bmatrix} F_0 & * \end{bmatrix} \cdot T^{-1}$ en A en B als in ISA bepaald, gaat dit over in:

$$\left(\begin{bmatrix} A_j & * & * \\ C_j & * & * \\ 0 & S_j & * \end{bmatrix} + \begin{bmatrix} B_j \\ D_j \\ 0 \end{bmatrix} \begin{bmatrix} F_0 & 0 & 0 \end{bmatrix} \right) \cdot \text{im } I_{n_j} \subset \text{im } I_{n_j}$$

waaruit:

$\text{im } A_j + \text{im } B_j F_0 \subset X_{n_j}$ hetgeen triviaal is.

$\text{im } C_j + \text{im } D_j F_0 = 0$, ofwel $D_j \cdot F_0 = -C_j$.

Uit de constructie van ISA volgt, dat $\text{im } C_j \subset \text{im } D_j$ zodat zo'n F_0 steeds gevonden kan worden. D_j behoeft echter geen volle rang te hebben zodat direct oplossen op bezwaar kan stuiten. Dit kan worden opgelost door eerst een kolomsamendrukking op D_j uit te voeren, als volgt:

outline

Aangezien (1) in de regel geen eenduidige oplossing heeft omdat D_j meestal geen volle rang heeft is directe berekening van F_0 door het direct oplossen van de lineaire vergelijking onmogelijk en dient een andere weg te worden bewandeld.

De bepaling van F_0 is nu als volgt:

- a. pas SVD toe op D_j , d.w.z. bepaal U ($p_j \times p_j$) en V^* ($m \times m$)

en S ($p_j \times m$) zodanig dat:

$$D_j = U \cdot S \cdot V^* \quad \text{met } S = \begin{bmatrix} S_a & 0 \\ 0 & 0 \end{bmatrix} \quad (2)$$

waarbij S_a slechts elementen ongelijk nul heeft op de diagonaal, nl. de singular values ongelijk nul.

- b. Vergelijking (1) gaan nu over in

$$U \cdot S \cdot V^* \cdot F_0 = -C_j \quad \text{ofwel} \quad \begin{bmatrix} S_a & 0 \\ 0 & 0 \end{bmatrix} \cdot V^* \cdot F_0 = -U^* \cdot C_j \quad (3)$$

Uit de constructie van ISA volgt dat $\text{im } C_j \subset \text{im } D_j$ zodat in het bijzonder geldt:

$$-U^* C_j = \begin{bmatrix} C_1 \\ 0 \end{bmatrix} \quad (4)$$

- c. Nu is F_1 eenduidig bepaald uit: $F_1 = S_a^{-1} \cdot C_1$ (5)

- d. De matrix F_0 volgt nu uit:

$$F_0 = V \cdot \begin{bmatrix} F_1 \\ 0 \end{bmatrix} \quad (6)$$

Hierbij zijn de nulelementen in principe willekeurig maar hier nul gekozen omdat we een feedbackmatrix in de regel zo simpel mogelijk willen houden.

- e. F zelf volgt nu uit:

$$F = \begin{bmatrix} F_0 & 0 \end{bmatrix} \cdot T^{-1} \quad (7)$$

opmerking: Het algoritme (ISA) kan stoppen met $cr = n_j$ in welk geval $\text{im } C_j \subset \text{im } D_j$ en bovenstaande algoritme is toe te passen. (ISA) kan echter ook stoppen met $n_j = 0$. Het resultaat is dan $V^* = (0)$. Duidelijk is dan dat we voor $F \in \underline{F}(V^*)$ iedere matrix kunnen kiezen. In het bijzonder is $F = 0$ een mogelijkheid.

verantwoording

Het hierboven beschreven algoritme ter bepaling van \underline{V}^* ker H komt uit /3/, algoritme 2.

III, B. Implementatie van ACSA.

Berekend wordt een orthogonale transformatie T_j , en getallen n_0 en n_j zodanig dat de kolommen n_j+1 tem. n_0 van T_j de ruimte \underline{R}_j opspannen. Bewezen wordt, dat \underline{R}_j voldoet aan het algoritme van Wonham:

$$\underline{R}_1 = (0) \quad (1)$$

$$\underline{R}_j = \underline{K} \cap (A \cdot \underline{R}_{j-1} + \text{im } B) \quad j=2,3,\dots$$

met $\underline{K} = \text{im } D = \underline{V}^* \text{ker } C$.

(Zie opmerking bij de vorige paragraaf over 'voldoen').

Als het algoritme stopt hebben we een orthogonale matrix $T_{\underline{x}}$, en getallen n_0 en $n_{\underline{x}}$ en spannen de $n_{\underline{x}}+1$ tem. n_0 de kolommen van $T_{\underline{x}}$ de ruimte $\underline{R}_{\underline{x}}$ op. $\underline{R}_{\underline{x}}$ voldoet aan (1) en is daarmee een goede benadering voor $\underline{R}^* \text{ker } C$.

outline van het algoritme

- a. Gegeven A ($n \times n$)

$$B$$
 ($n \times m$) ofwel: $P = \begin{bmatrix} A & B \\ D^* & 0 \end{bmatrix}$

$$D$$
 ($n \times q$)

- b. Bepaal T_0 ($n \times n$) en S_0 ($n_0 \times q$) zodanig dat:

$$T_0^* \cdot D = \begin{bmatrix} S_0 \\ 0 \end{bmatrix}$$

waarbij n_0 de rijrang is van D .

- c. Stop indien $n_0 = 0$. (dan is $\underline{R}_{\underline{x}} = \underline{V}^* = 0$)

- d. Pas B aan:

$$T_0^* \cdot B = \begin{bmatrix} B_1^1 \\ B_0^1 \\ B_0^2 \\ B_0^3 \end{bmatrix}$$

- e. Bepaal V ($m \times m$) en B_0^{22} ($n-n_0, cr$) zodanig dat:

$$B_0^{22} \cdot V = \begin{bmatrix} 0 & B_0^{22} \end{bmatrix}$$

waarbij cr de kolomrang is van B_0^{22} .

- f. Stop indien $cr = m$ (Dan is $\text{im } B \cap \text{im } D = 0$ en dus $\underline{R}_{\underline{x}} = 0$).

- g. Pas B_0^1 aan:

$$B_0^1 \cdot V = \begin{bmatrix} B_0^{11} & B_0^{12} \end{bmatrix}$$

- h. Bepaal U ($n_0 \times n_0$) en S_1 ($m \times rr$) zodanig dat:

$$B_0^{11} \cdot U = \begin{bmatrix} 0 & S_1 \end{bmatrix}$$

waarbij rr de rijrang is van B_0^{11} .

(merk op dat hier in feite een rijsamendrukking plaats vindt:

$$U^* \cdot B_0^{11} = \begin{bmatrix} 0 \\ S_1^* \end{bmatrix})$$

i. Stop indien $rr = 0$ (dan is $\text{im } B \cap \text{im } D = 0$ en dus $R_{\infty} = 0$)

j. Herbenoemen:

$$T_1 = T_0 \cdot \text{diag}(U, I_{p_0}) \quad (\text{met } p_0 = n - n_0)$$

$$\begin{bmatrix} T_1 & 0 \\ 0 & I_q \end{bmatrix}^{-1} \cdot P \cdot \begin{bmatrix} T_1 & 0 \\ 0 & I_m \end{bmatrix} = \begin{bmatrix} T_1^{-1} A T_1 & T_1^{-1} B \\ -S_0^* & 0 \end{bmatrix} =$$

$$\begin{bmatrix} T_1^{-1} A T_1 & T_1^{-1} B \\ -S_0^* & 0 \end{bmatrix} \cdot \begin{bmatrix} I_n & 0 \\ 0 & V^* \end{bmatrix} = \begin{bmatrix} A_1 & B_1 & * & 0 & * \\ * & * & S_1 & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ S_0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} I_n & 0 \\ 0 & V^* \end{bmatrix}$$

$$n_1 = n - n_0$$

$$j = 2$$

k. Bepaal U ($n_{j-1} \times n_{j-1}$) en S_j ($n_{j-1} \times rr$) zodanig dat:

$$B_{j-1}^* \cdot U = \begin{bmatrix} 0 & S_j \end{bmatrix}$$

waarbij rr de rijrang is van B_{j-1} .

(zie de opmerking bij stap h.)

l. stop indien $rr = 0$ (resultaat is gelijk aan dat van de vorige stap).

m. stop indien $rr = n_{j-1}$ (resultaat valt samen met $\text{im } D$, de laatste transformatie U heeft geen wezenlijke invloed meer).

n. Herbenoemen:

$$T_j = T_{j-1} \cdot \text{diag}(U, I_{n-n_j})$$

$$n_j = n_{j-1} - rr$$

$$\begin{bmatrix} T_j & 0 \\ 0 & I_q \end{bmatrix}^{-1} \cdot P \cdot \begin{bmatrix} T_j & 0 \\ 0 & I_m \end{bmatrix} =$$

$$\begin{bmatrix} U^* & 0 \\ 0 & I \end{bmatrix} \cdot \begin{bmatrix} A_{j-1} & B_{j-1} & 0 & 0 & \cdot & 0 & * \\ * & * & S_{j-1}^* & 0 & \cdot & 0 & * \\ * & * & * & S_{j-2}^* & \cdot & 0 & * \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & * \\ * & * & * & * & \cdot & S_2^* & * \\ * & * & * & * & \cdot & * & * \\ S_0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} U & 0 \\ 0 & I \end{bmatrix}$$

$$\begin{bmatrix} U^* A_{j-1} U & U^* B_{j-1} & 0 & * & * \\ * & * & S_j^* & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ S_0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} U^* A_{j-1} U & 0 & * & * \\ * & S_j^* & 0 & * \\ * & * & S_j^* & * \\ * & * & * & * \\ S_0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} U & 0 \\ 0 & I \end{bmatrix}$$

$$= \begin{bmatrix} A_j & B_j & 0 & 0 & * \\ * & * & S_j^* & 0 & * \\ * & * & * & S_j^* & * \\ * & * & * & * & * \\ S_0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot x$$

o. $j = j + 1$

herhaal vanaf stap k.

bewijs van deze implementatie

We bewijzen dat de implementatie equivalent is met het volgende algoritme:

$$R_0 = \text{im } D$$

$$R_1 = R_0 \cap \text{im } B$$

(2)

$$R_j = R_{j-1} + (R_0 \cap A R_{j-1})$$

Het is niet moeilijk in te zien, dat (2) equivalent is met (1).

We gebruiken de volgende notatie:

$$I_n = \text{sp}(e_1, \dots, e_n) \quad \text{en} \quad I_n^m = \text{sp}(e_{m+1}, \dots, e_n) \quad \text{en daarmee} \quad S_0 =$$

$$T_0 \text{ im } I_{n_0} = \text{sp}(t_1, \dots, t_{n_0}) \quad \text{en} \quad S_j = T_j \text{ im } I_{n_0}^j = \text{sp}(t_{n_{j-1}+1}, \dots, t_{n_0}).$$

We bewijzen nu dat de zo gedefinieerde S_j voldoet aan (2).

a. $S_0 = \text{im } D$

Dit is direct duidelijk uit de stappen a en b.

b. $S_1 = \text{im } D \cap \text{im } B$.

Ofwel:

$$T_1 \text{ im } I_{n_0}^{n_1} = T_0 \text{ im } I_{n_0} \cap \text{im } B$$

Links vermenigvuldigen met T_1^{-1} geeft:

$$\text{im } I_{n_0}^{n_1} = \text{diag}(U^*, I) \text{ im } I_{n_0} \cap T_1^{-1} \text{ im } B = \text{im } I_{n_0} \cap T_1^{-1} \text{ im } B$$

We werken $T_1^{-1} B$ even verder uit:

$$T_1^{-1} \cdot B = \begin{bmatrix} U^* & 0 \\ 0 & I \end{bmatrix} \cdot T_0 \cdot B = \begin{bmatrix} U^* & 0 \\ 0 & I \end{bmatrix} \cdot \begin{bmatrix} B_0^1 \\ B_0^2 \end{bmatrix} = \begin{bmatrix} U^* & 0 \\ 0 & I \end{bmatrix} \cdot \begin{bmatrix} B_0^1 \\ B_0^2 \end{bmatrix} \cdot V \cdot V^*$$

$$\begin{bmatrix} U^* & 0 \\ 0 & I \end{bmatrix} \cdot \begin{bmatrix} B_0^{11} & B_0^{12} \\ 0 & B_0^{22} \end{bmatrix} \cdot V^* = \begin{bmatrix} U^* B_0^{11} & U^* B_0^{12} \\ 0 & B_0^{22} \end{bmatrix} \cdot V^* =$$

$$\begin{bmatrix} 0 & * \\ S_j^* & U^* B_0^{12} \\ 0 & B_0^{22} \end{bmatrix} \cdot V^* = \begin{bmatrix} 0 & * \\ S_j^* & * \\ 0 & * \end{bmatrix} \cdot V^*$$

Nu is direct duidelijk dat $\text{im } I_{n_0} \cap \text{im } T_1^{-1} B = \text{im } I_{n_0}^{n_1}$.

Merk op dat de resterende transformatie V^* geen wezenlijke invloed heeft op $\text{im } T_1^{-1}B$.

In de rest van het algoritme worden nog slechts transformaties U bepaald, die enkel de eerste n_j kolommen van de transformatie T_j beroeren. Hieruit volgt dat de laatste kolommen vanaf kolom nummer n_1+1 onveranderd blijven. We kunnen nu concluderen dat in de rest van het algoritme de kolommen n_1+1 t.m. n_0 steeds $\text{im } B \cap \text{im } D$ zullen opspannen. \underline{S}_j is dus steeds gelijk aan \underline{S}_1 + 'nogwat'.

$$c. \underline{S}_j = \underline{S}_1 + (\underline{S}_0 \cap A \cdot \underline{S}_{j-1})$$

Uitgewerkt hebben we:

$$T_j \cdot \text{im } I_{n_0}^{n_j} = T_1 \cdot \text{im } I_{n_0}^{n_1} + (T_0 \cdot \text{im } I_{n_0} \cap A \cdot T_{j-1} \cdot \text{im } I_{n_0}^{n_{j-1}})$$

Links vermenigvuldigen met T_j^{-1} geeft:

$$\text{im } I_{n_0}^{n_j} = \text{im } I_{n_0}^{n_1} + (\text{im } I_{n_0} \cap T_j^{-1} \cdot A \cdot T_{j-1} \cdot \text{im } I_{n_0}^{n_{j-1}})$$

waarbij $T_j^{-1} \cdot T \cdot \text{im } I_{n_0}^{n_1} = \text{im } I_{n_0}^{n_1}$ vanwege de opmerking bij b. en

$T_j^{-1} \cdot T_0 \cdot \text{im } I_{n_0} = \text{im } I_{n_0}$ vanwege soortgelijke beschouwingen).

Uitwerken geeft:

$$\text{im } I_{n_0}^{n_j} = \text{im } I_{n_0} \cap \begin{bmatrix} U^* & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{j-1} & B_{j-1} & 0 & * \\ * & * & S & * \\ * & * & * & * \end{bmatrix}$$

M.b.v. de uitwerking in stap n vinden we:

$$\text{im } I_{n_0}^{n_j} = \text{im } I_{n_0} \cap \begin{bmatrix} U^* A_{j-1} & U^* B_{j-1} & 0 & * \\ * & * & S & * \\ * & * & * & * \end{bmatrix} \cdot \text{im } I_{n_0}^{n_{j-1}}$$

waarbij uit:

$$U^* \cdot B_{j-1} = \begin{bmatrix} 0 \\ S_j^* \end{bmatrix}$$

direct het gestelde volgt.

verantwoording

Het hierboven vermelde algoritme ter bepaling van $\underline{R}^* \text{ker } H$ is een modificatie van algoritme 3* uit /3/. Omdat de bepaling van een feedbackmatrix F zodanig dat $\underline{R}^* \text{ker } H$ $(A + BF)$ -invariant is, hier niet nodig is, kon het worden vereenvoudigd.

III.C. Bepaling van een $F \in F_p(\underline{R}^* \text{ker } H)$.

De bedoeling is een F te bepalen zodanig dat:

$$(A + B \cdot F) \underline{R}^* \subset \underline{R}^* \quad (1)$$

$$\text{spec}((A + B \cdot F)|_{\underline{R}^*}) \text{ specificiceerbaar} \quad (2)$$

$$\text{spec}((A + B \cdot F)(\text{mod } \underline{R}^*)) \text{ specificiceerbaar} \quad (3)$$

de derde eis is slechts mogelijk als (A, B) regelbaar is.

Veronderstel F van de vorm:

$$F = F_a + F_b \quad \text{met } F_a \in F(\underline{V}^*) \subset F(\underline{R}^*) \quad (4)$$

Pas verder een basistransformatie T toe, zodat $T^{-1}x = (r_1 \ r_2)^t$ met $r_1 \in \underline{R}^*$ en $r_2 \in \underline{X} \setminus \underline{R}^*$.

Dan is:

$$T^{-1} \cdot (A + B \cdot F_a) \cdot T = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \quad \text{en } T^{-1} \cdot B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \quad (5)$$

Waarbij het element A_{21} nul is vanwege de keuze van F_a .

Door de keuze van F_a is aan de eerst eis (1) voldaan. Door geschikte keuze van F_b moeten de beide andere eisen worden voldaan en wel zo dat de invariantieeis (1) niet verloren gaat.

Hiertoe passen we kolomsamendrukking toe op matrix B_2 , d.w.z.:

$$\begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \cdot Q = \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{bmatrix} \quad (6)$$

Veronderstel verder:

$$Q^* \cdot F_b \cdot T = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \quad (7)$$

Dan blijft aan eis (1) voldaan, d.e.s.d.a. $B_{22}F_{21} = 0$.

Door de kolom samendrukking van B_2 heeft B_{22} volle kolomrang en is F_{21} eenduidig bepaald uit $B_{22}F_{21} = 0$ en dus $F_{21} = 0$.

We kunnen nu drie gevallen onderscheiden, noem daarbij de kolomrang van B_2 (en dus van B_{22}) cr.

i. cr = 0.

We vinden:

$$T^{-1} \cdot (A + B \cdot F) \cdot T = \begin{bmatrix} A_{11} + B_{11} \cdot F_{11} & A_{12} + B_{11}F_{12} \\ 0 & A_{22} \end{bmatrix} \quad (8)$$

M.b.v. de matrices A_{11} en $B_{11} = B_1$ is een matrix F_{11} te bepalen zodanig dat $\text{spec}(A_{11} + B_{11}F_{11})$ gelijk is aan de gewenste polen. Dit duidt op poolplaatsbaarheid in $\underline{R}^* \text{ker } H$. De polen in $\underline{X} \setminus \underline{R}^* \text{ker } H$ liggen echter vast en kunnen niet gespecificeerd worden. Deze situatie treedt slechts op als (A, B) niet regelbaar is.

ii. $cr = m$ ($m = \text{dimensie input}$)

De situatie is nu:

$$T^{-1} \cdot (A + B \cdot F) \cdot T = \begin{bmatrix} A_{11} & A_{12} + B_{12} F_{22} \\ 0 & A_{22} + B_{22} F_{22} \end{bmatrix} \quad (9)$$

Deze situatie is in tegenspraak met de gestelde eisen op de deelruimte \underline{R}^* : hier is geen poolplaatsing mogelijk in \underline{R}^* !!!

Deze situatie zal dan ook nimmer optreden.

iii. $0 < cr < m$

De meest algemene situatie:

$$T^{-1} \cdot (A + B \cdot F) \cdot T = \begin{bmatrix} A_{11} + B_{11} F_{11} & A_{12} + B_{11} F_{12} + B_{12} F_{22} \\ 0 & A_{22} + B_{22} F_{22} \end{bmatrix} \quad (10)$$

We zien dat hier poolplaatsing mogelijk is in de gehele ruimte (mits (A, B) regelbaar):

spec $(A_{11} + B_{11} F_{11})$ specificeert de polen in \underline{R}^* ker H en

spec $(A_{22} + B_{22} F_{22})$ specificeert de polen in $\underline{X} \setminus \underline{R}^*$ ker H.

opmerkingen

i. poolplaatsing in \underline{R}^* ker H is altijd mogelijk (zo is \underline{R}^* immers gedefinieerd). Uit bovenstaande constructie volgt, dat (A_{11}, B_1) altijd regelbaar is. In combinatie met eis (1) volgt dat (A_{11}, B_{11}) altijd regelbaar is.

ii. poolplaatsing in $\underline{X} \setminus \underline{R}^*$ ker H is slechts mogelijk indien (A, B) regelbaar is. Alleen dan volgt uit de constructie dat (A_{22}, B_2) regelbaar is.

iii. Indien (A, B) regelbaar is en het volledige spectrum gespecificeerd wordt, d.w.z. spec $(A + B \cdot F)$ is gelijk aan een vooraf gekozen spectrum, is een extra voorwaarde op het gewenste spectrum nodig. Het gewenste spectrum dient opgesplitst te kunnen worden in twee deelspectra, die beide symmetrisch zijn. Immers, voor poolplaatsing is nodig, dat een spectrum symmetrisch is, d.w.z. dat met z ook de geadjungeerde z^* in het spectrum voorkomt.

Indien beide deelspectra een even aantal polen (moeten) bevatten is er geen extra eis nodig, maar indien minstens een van deze deelspectra een oneven aantal polen bevat is het nodig van het gehele spectrum te eisen dat zo'n splitsing gemaakt kan worden. Bijvoorbeeld een verzameling van polen als $(1 + i, 1 - i, i, -i)$ kan niet worden opgedeeld in twee symmetrische deelspectra met resp. 1 en 3 elementen. Een voldoende voorwaarde voor het (gehele) gewenste spectrum is de eis dat het minstens één reeel element bevat. Deze voorwaarde is echter niet noodzakelijk.

constructie

Bovenstaande geeft aanleiding tot de volgende constructie:

i. Pas de transformatie toe als in (5)

ii. Pas kolom samendrukking toe op B_2 , d.w.z. bepaal Q en B_{22} zodanig dat: $B_2 \cdot Q = \begin{bmatrix} 0 & B_{22} \end{bmatrix}$

Bepaal B_{11} uit $B_1 \cdot Q = \begin{bmatrix} B_{11} & B_{12} \end{bmatrix}$

iii. Pas poolplaatsingsalgoritme toe met A_{11} en B_{11} , dwz.

bepaal een F_{11} zodanig dat $(A_{11} + B_{11} F_{11})$ gewenste polen heeft.

iv. Indien het gehele spectrum gespecificeerd wordt:

Pas poolplaatsingsalgoritme toe met A_{22} en B_2 , dwz.

bepaal een F_2 zodanig dat $(A_{22} + B_2 F_2)$ gewenste polen heeft.

Kies anders $F_2 = 0$.

v. F volgt nu uit:

$$F = F_a + \begin{bmatrix} Q \cdot \begin{bmatrix} F_{11} \\ 0 \end{bmatrix} \\ F_2 \end{bmatrix} \cdot T^{-1}$$

Ter controle:

$$T^{-1} \cdot (A + B \cdot F) \cdot T = T^{-1} \cdot (A + B \cdot F_a) \cdot T + T^{-1} \cdot B \cdot F_b \cdot T =$$

$$\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \cdot Q \cdot \begin{bmatrix} F_{11} \\ 0 \end{bmatrix} \cdot F_2 =$$

$$\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} + \begin{bmatrix} B_1 \cdot Q \cdot \begin{bmatrix} F_{11} \\ 0 \end{bmatrix} \\ B_2 \cdot Q \cdot \begin{bmatrix} F_{11} \\ 0 \end{bmatrix} \end{bmatrix} + \begin{bmatrix} B_1 F_2 \\ B_2 F_2 \end{bmatrix} =$$

$$\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} + \begin{bmatrix} B_{11} F_{11} & B_1 F_2 \\ 0 & B_2 F_2 \end{bmatrix} =$$

$$\begin{bmatrix} A_{11} + B_{11} F_{11} & A_{12} + B_1 F_2 \\ 0 & A_{22} + B_2 F_2 \end{bmatrix}$$

waarmee aangetoond is dat deze constructie aan de eisen (1), (2) en (eventueel) (3) voldoet.

III.D. Bepaling van een $L \in L_p(N^* \text{ in } G)$.

De bedoeling is nu een L te bepalen zodat:

$$(A + L \cdot C) \cdot N^* \subset N^* \quad (1)$$

$$\text{spec}((A + L \cdot C) \mid N^*) \text{ specificiceerbaar} \quad (2)$$

$$\text{spec}((A + L \cdot C) \pmod{N^*}) \text{ specificiceerbaar} \quad (3)$$

waarbij de tweede eis (2) slechts mogelijk is als (A, C) waarneembaar is.

We passen dezelfde methode toe als in III.C. Veronderstel

$$L = L_a + L_b \text{ met } L_a \in L(\underline{S}^*) \subset L(N^*) \quad (4)$$

en passen wederom een transformatie toe, zodat $T^{-1}x = (n_1 \ n_2)^t$ met $n_1 \in N^*$ en $n_2 \in \underline{X} \setminus N^*$.

Dan is weer:

$$T^{-1}(A + L_a \cdot C) \cdot T = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \text{ en } C \cdot T = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \quad (5)$$

Rij-samendrukking van matrix C_1 levert nu:

$$Q \cdot \begin{bmatrix} C_1 & C_2 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ 0 & C_{22} \end{bmatrix} \quad (6)$$

Veronderstel verder:

$$T^{-1} \cdot L_b \cdot Q^* = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \quad (7)$$

Dan volgt uit het voldaan zijn aan eis (1) dat $L_{21} = 0$.

Afhankelijk van de rijrang rr van de matrix C_1 volgen weer drie mogelijkheden:

i. $rr = 0$

$$T^{-1} \cdot (A + L \cdot C) \cdot T = \begin{bmatrix} A_{11} & A_{12} + L_{12}C_{22} \\ 0 & A_{21} + L_{22}C_{22} \end{bmatrix}$$

Poolplaatsbaarheid is mogelijk in $\underline{X} \setminus N^*$, maar niet in N^* . Deze situatie treedt slechts op als (A, C) niet waarneembaar is.

ii. $rr = p$ ($p = \text{dimensie output}$)

Deze situatie treedt nimmer op (vergelijk met de situatie $cr = m$ in III.C.)

iii. $0 < rr < p$

$$T^{-1} \cdot (A + L \cdot C) \cdot T = \begin{bmatrix} A_{11} + L_{11}C_{11} & A_{12} + L_{11}C_{12} + L_{12}C_{22} \\ 0 & A_{22} + L_{22}C_{22} \end{bmatrix}$$

$\text{spec}(A_{11} + L_{11}C_{11})$ specificiceert de polen in $N^* \text{ in } G$

$\text{spec}(A_{22} + L_{22}C_{22})$ specificiceert de polen in $\underline{X} \setminus N^* \text{ in } G$

In de laatste situatie is poolplaatsbaarheid mogelijk in de hele ruimte \underline{X} mits (A, C) waarneembaar is.

Hier gelden weer dezelfde opmerkingen als in III.E.

constructie.

De constructie is nu als volgt:

i. Pas de transformatie toe als in (5)

ii. Pas rij samendrukking toe op C_1 , d.w.z. bepaal Q en C

zodat:

$$Q \cdot C_1 = \begin{bmatrix} C_{11} \\ 0 \end{bmatrix}$$

Bepaal C_{22} uit:

$$Q \cdot C_2 = \begin{bmatrix} C_{12} \\ C_{22} \end{bmatrix}$$

iii. Pas poolplaatsingsalgoritme toe met A_{22} en C_{22} , d.w.z.

bepaal L_{22} zodanig dat $(A_{22} + L_{22}C_{22})$ gewenste polen heeft.

iv. Indien het gehele spectrum gespecificeerd wordt:

Pas poolplaatsingsalgoritme toe met A_{11} en C_1 , d.w.z.

bepaal L_1 zodanig, dat $(A_{11} + L_1C_1)$ gewenste polen heeft.

Kies anders $L_1 = 0$.

v. L volgt nu uit:

$$L = L_a + T \cdot \begin{bmatrix} L_1 & L_{22} \\ 0 & L_{22} \end{bmatrix} \cdot Q$$

En inderdaad geldt nu:

$$T^{-1} \cdot (A + L \cdot C) \cdot T = T^{-1} \cdot (A + L_a \cdot C) \cdot T + T^{-1} \cdot L_b \cdot C \cdot T =$$

$$\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{21} \end{bmatrix} + \begin{bmatrix} L_1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} C_{11} & C_{12} \\ 0 & C_{22} \end{bmatrix} =$$

$$\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} + \begin{bmatrix} L_1C_1 & L_1C_2 \\ 0 & L_{22}C_{22} \end{bmatrix} \cdot Q \cdot C_1 =$$

$$\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} + \begin{bmatrix} L_1C_1 & L_1C_2 \\ 0 & L_{22}C_{22} \end{bmatrix} =$$

$$\begin{bmatrix} A_{11} + L_1C_1 & A_{12} + L_1C_2 \\ 0 & A_{22} + L_{22}C_{22} \end{bmatrix}$$

opmerking

In feite zijn de problemen III.C. en III.D. elkaars duale. Dit volgt het snelst door in het voorgaande de transformatie te vervangen door $T^{-1}x = (n_1 \ n_2)^t$ met $n_1 \in \underline{X} \setminus \underline{N}^{\#}$ en $n_2 \in \underline{N}^{\#}$. Hieruit volgt dat men kan volstaan met één algoritme, bijvoorbeeld dat beschreven in III.C. (waarbij A vervangen is door A^t , B door C^t en T door T^t zodat de L dan volgt uit $L = F^t$). Om programmeer-technische redenen is echter gekozen voor het apart implementeren van beide algoritmen. In het LINDIST-pakket wordt namelijk de basistransformatie $T^{-1}x = (n_1 \ n_2)^t$ gebruikt met $n_1 \in \underline{N}^{\#}$ en $n_2 \in \underline{X} \setminus \underline{N}^{\#}$.

III.E. Bepaling van een $K \in K(V, S)$.

De bedoeling is het bepalen van een matrix K waarvoor
 $(A + B \cdot K \cdot C) \underline{S} \subset \underline{V}$ (1)

Een dergelijke matrix bestaat als voldaan is aan:

$$\underline{S} \subset \underline{V} \quad (2)$$

en houdt verband met zg. (A,B,C)-invariante deelruimten (zie /5/ en /6/). Hier zullen dergelijke matrices worden gebruikt bij het bepalen van oplossingen bij DDPM en DDPMPP.

Uit onderstaande constructie van zo'n K volgt dat (2) een voldoende voorwaarde is wil zo'n matrix bestaan. De constructie is afgeleid uit lemma 2.2 in /5/.

methode

We kiezen eerst een $L \in \underline{L}(\underline{S})$, zodat:

$$(A + L \cdot C) \underline{S} \subset \underline{S} \quad (3)$$

en bepalen vervolgens een matrix L_0 die voldoet aan

$$\begin{aligned} \text{im } (L - L_0) &\subset \text{im } B \\ \text{im } L_0 &\subset \underline{V} \end{aligned} \quad (4)$$

Uit (4) volgt dat $B \cdot K = L - L_0$ oplosbaar is (d.i. zo'n K is te bepalen) en geldt:

$$\begin{aligned} (A + B \cdot K \cdot C) \cdot \underline{S} &= (A + (L - L_0) \cdot C) \cdot \underline{S} = (A + L \cdot C) \cdot \underline{S} - L_0 \cdot C \cdot \underline{S} \\ &\subset \underline{S} + \underline{V} \subset \underline{V} \end{aligned}$$

zodat inderdaad de gewenste K gevonden is.

We dienen nog na te gaan of zo'n matrix L_0 inderdaad bestaat. Stel eerst dat L_0 inderdaad altijd gevonden kan worden.

De procedure is dan als volgt:

i. Pas rijsamendrukking toe op de matrix B, d.w.z. bepaal $U^{\#}$

en B_1 zodat:

$$U^{\#} \cdot B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \quad (5)$$

ii. Veronderstel verder:

$$U^{\#} \cdot L = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} \quad (6)$$

iii. Als $L_2 = 0$ kies dan $L_0 = 0$.

Is $L_2 \neq 0$, bepaal L_0 dan zodanig dat voldaan is aan:

$$U^{\#} \cdot L_0 = \begin{bmatrix} * \\ L_2 \end{bmatrix} \quad \text{en} \quad \text{im } L_0 \subset \underline{V} \quad (7)$$

als volgt:

Los de matrix Y op uit:

$$V_2 \cdot Y = L_2 \quad (8)$$

waarbij:

$$U^{\#} \cdot V = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \quad (9)$$

Hier is V de matrix met als kolommen de basisvectoren van de ruimte \underline{V} .

L_0 volgt dan uit:

$$L_0 = V \cdot Y \quad (10)$$

(immers: (10) impliceert $\text{im } L_0 \in \underline{V}$ en bovendien is

$$U^{\#} \cdot L_0 = U^{\#} \cdot V \cdot Y = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} Y = \begin{bmatrix} V_1 Y \\ V_2 Y \end{bmatrix})$$

iv. Los K nu op uit

$$B \cdot K = L - L_0 \quad (11)$$

(merk op, dat uit (11) volgt, door links te vermenigvuldigen met $U^{\#}$, dat:

$$\begin{bmatrix} B_1 \\ 0 \end{bmatrix} K = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} - \begin{bmatrix} \# \\ L_2 \end{bmatrix} = \begin{bmatrix} \# \\ 0 \end{bmatrix}$$

en inderdaad een oplossing K bestaat)

Blijft nog over de vraag of een dergelijke L_0 altijd gevonden kan worden, d.w.z. of een matrix Y bepaalt kan worden als in (8).

Welnu, er geldt:

$$A \cdot \underline{S} \subset A \cdot \underline{V} \subset \underline{V} + \text{im } B \quad (12)$$

en ook:

$$(A + L \cdot C) \cdot \underline{S} = A \cdot \underline{S} + L \cdot C \cdot \underline{S} \subset \underline{S} \quad (13)$$

Uit (13) in combinatie met (12) volgt:

$$L \cdot C \cdot \underline{S} \subset \text{im } B + \underline{V} \text{ ofwel}$$

$$\text{im } L \subset \text{im } B + \underline{V}$$

Indien nu $\text{im } L \subset \text{im } B$ verkeren we in de situatie waarin $L_0 = 0$ gekozen kan worden en indien $\text{im } L \subset \text{im } B + \underline{V}$ volgt dat de matrix Y als in (8) bestaat, omdat $\text{im } L_2 = \text{im } (L(\text{mod im } B)) \subset \underline{V}(\text{mod im } B) = \text{im } V_2$

De bepaling van de matrix Y in (8) en de matrix K in (11) gaat op dezelfde manier als besproken is in III.A. bij de bepaling van $F \in \underline{F}(\underline{V}^{\#})$.

III.F. Poolplaatsingsalgoritme

Gegeven dat (A, B) regelbaar is, komt een poolplaatsingsalgoritme neer op het bepalen van een matrix F , zodanig dat

$$\det (I s - A - B \cdot F) = p(s) \quad (1)$$

waarbij $p(s)$ een polynoom is met wortels juist de opgegeven polen.

Gegeven (A, C) waarneembaar, dient een matrix L bepaalt te worden zodat:

$$\det (I s - A - L \cdot C) = p(s) \quad (2)$$

Wegens dualiteit, immers $\det (I s - A - L \cdot C) = \det (I s - A^t - C^t L^t)$ is het voldoende probleem (1) op te lossen.

De methode is gesplitst in twee delen:

- Bepaal een $b \in \text{im } B$ en een matrix F_1 zodat $(A + B \cdot F_1, b)$ regelbaar is.
- Bepaal een vector k^t , zodat $\det (I s - A - B \cdot F_1 - b \cdot k^t) = p(s)$. Het oplossen van i. is heel eenvoudig. Er geldt namelijk, dat $(A + B \cdot F, b)$ generiek regelbaar is voor F . $(0 \neq b \in \text{im } B)$. (2)

Voor het begrip genericiteit alsmede voor een bewijs van (2), zie bijvoorbeeld /1/.

De procedure is nu als volgt:

- Ga na of er een kolom b van B is, waarvoor (A, b) regelbaar is.
- Lukt i.a. niet, neem dan een kolom b van B , ongelijk nul, en bepaal F_1 m.b.v. een random-generator. (Uit (2) volgt dat zo'n random F_1 met kans 1 voldoet).

Het oplossen van ii. betekent het oplossen van k^t uit een vergelijking van de vorm:

$$\det (I s - \bar{A} - b \cdot k^t) = s^n + p_{n-1} s^{n-1} + \dots + p_0 \quad (3)$$

Stel nu $F = \bar{A} + b \cdot k^t$ en tel onderstaande vergelijkingen op:

$$\begin{array}{lcl} I & = & I \\ F & = & \bar{A} + b \cdot k^t \\ F^2 & = & \bar{A}^2 + \bar{A} b \cdot k^t + b \cdot k^t F \\ F^3 & = & \bar{A}^3 + \bar{A}^2 b \cdot k^t + \bar{A} b \cdot k^t F + b \cdot k^t F^2 \\ & \vdots & \\ F^n & = & \bar{A}^n + \bar{A}^{n-1} b \cdot k^t + \dots + b \cdot k^t F^{n-1} \end{array} \quad \begin{array}{l} \times p_0 \\ \times p_1 \\ \times p_2 \\ \times p_3 \\ \vdots \\ \times 1 \end{array} +$$

$$p(F) = p(\bar{A}) +$$

$$\begin{bmatrix} b, \bar{A} b, \dots, \bar{A}^{n-1} b \end{bmatrix} \cdot \begin{bmatrix} k^t F^{n-1} + p_{n-1} F^{n-2} + \dots + p_1 k^t \\ \vdots \\ k^t F + p_{n-1} k^t \\ k^t \end{bmatrix}$$

Toepassing van de stelling van Cayley-Hamilton leert ons dat

$$p(F) = 0 \quad (4)$$

zodat uit het voorgaande volgt, samen met het feit, dat regelbaarheid van (\bar{A}, b) impliceert dat de regelbaarheidsmatrix regulier en dus inverteerbaar is, dat:

$$k^t = [0 \dots 0 \ 1] \cdot [b, \bar{A} b, \dots, \bar{A}^{n-1} b]^t \cdot p(\bar{A}) \quad (5)$$

Stel $e^t = [0 \dots 0 \ 1] \cdot [b, \bar{A} b, \dots, \bar{A}^{n-1} b]^t$, dan is $k^t = e^t p(\bar{A})$.

De methode is nu als volgt:

ii.a. los het stelsel:

$$[b, \bar{A} b, \dots, \bar{A}^{n-1} b]^t \cdot e = [0 \dots 0 \ 1]^t \quad (6)$$

op.

ii.b. bepaal k^t uit:

$$k^t = p_0 e^t + p_1 e^t \bar{A} + \dots + p_{n-1} e^t \bar{A}^{n-1} + e^t \bar{A}^n \quad (7)$$

In totaliteit komt het algoritme dus neer op:

i. Bepaal F_1 zodat voor een kolom b van B ($A + B F_1, b$) regelbaar is.

ii. los e^t op uit (6) en bepaal k^t uit (7) met $\bar{A} = A + B F_1$

iii. de gezochte F is nu gelijk aan:

$$F = F_1 + b k^t \quad (8)$$

hoofdstuk IV. STORINGSONTKOPPELINGSPROBLEMEN.

Bij de storingsontkoppelings-problemen, zoals hierna besproken zullen worden, wordt uitgegaan van het volgende systeem:

$$\begin{aligned}\dot{x} &= A x + B u + G d \\ y &= C x \\ z &= H x\end{aligned}\quad (*)$$

Hierbij is:

$x \in \underline{X} = \mathbb{R}^n$ de toestand
 $y \in \underline{Y} = \mathbb{R}^p$ de waarneming
 $u \in \underline{U} = \mathbb{R}^m$ de sturing
 $d \in \underline{D} = \mathbb{R}^q$ de verstoring
 $z \in \underline{Z} = \mathbb{R}^r$ de te regelen of te schatten variabele

IV.A. DDP: Disturbance Decoupling Problem.

Uitgaande van systeem (*) met $C = I$ (d.w.z. de gehele toestand kan worden waargenomen) wordt gezocht een feedbackregelaar van de vorm:

$$u = F \cdot x \quad (1)$$

zodanig dat de storingsen d geen invloed hebben op de te regelen variabelen z .

Wiskundig gezien komt dit neer op het bestaan van een matrix F zodanig dat de transferfunctie van het closed-loop-systeem nul is, d.w.z.:

$$H \cdot (I_s - A_F)^{-1} \cdot G = 0 \quad (2)$$

We hebben dus:

(DDP) Bestaat een F zodanig dat $H \cdot (I_s - A - B \cdot F)^{-1} \cdot G = 0$?

en:

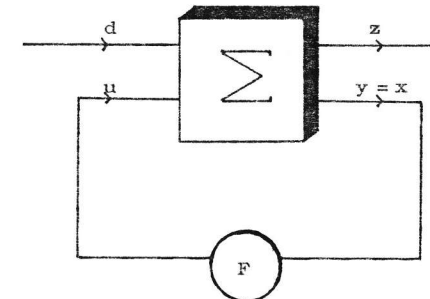
(DDPPP) Bestaat een F zoals in (DDP) terwijl bovendien de polen van $(A + B \cdot F)$ willekeurig te kiezen zijn. ?

Beide problemen zijn opgelost in /2/, waar nodige en voldoende voorwaarden zijn afgeleid, nl.:

(DDP) is oplosbaar desda $\text{im } G \subset \underline{V}^* \ker(H)$ (3)

(DDPPP) is oplosbaar desda $\text{im } G \subset \underline{R}^* \ker(H)$ (4)
en (A, B) regelbaar

Een oplossing voor (DDP) is een $F \in \underline{F}(\underline{V}^* \ker(H))$, een oplossing voor (DDPPP) is een $F \in \underline{F}_p(\underline{R}^* \ker(H))$ met $\text{spec}(A + B F)$ specificeerbaar.



bewijs van (3) en (4)

Noodzakelijkheid: zie /2/.

Voldoende:

Aangetoond wordt, dat de voorwaarden voldoende zijn wil de aangegeven oplossing inderdaad voldoen.

Pas een basistransformatie T toe, zodat $T^{-1}x = (v_1 \ v_2)^t$ met $v_1 \in \underline{V}^*$ en $v_2 \in \underline{X} \setminus \underline{V}^*$.

Toepassing van de feedback $u = Fx$ met $F \in \underline{F}(\underline{V}^*)$ geeft dan:

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} A_{11}+B_1F_1 & A_{12}+B_1F_2 \\ 0 & A_{22}+B_2F_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} G_1 \\ 0 \end{bmatrix} d$$

$$z = \begin{bmatrix} 0 & H_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = H_2 v_2 \quad (5)$$

Waarbij $G_2 = 0$ omdat $\text{im } G \subset \underline{V}^*$ (De eis)

en $H_1 = 0$ omdat $\underline{V}^* \subset \ker H$ (per definitie).

Hieruit volgt ogenblikkelijk dat $z = H_2 v_2$ en $\dot{v}_2 = (A_{22}+B_2F_2)v_2$ zodat z onafhankelijk is van d .

Onder de voorwaarde zoals gesteld in (3) is DDP oplosbaar en een mogelijke oplossing is $F \in \underline{F}(\underline{V}^*)$.

Het bewijs voor (4) gaat analoog. Voor de constructie van een F zodat $F \in \underline{F}(\underline{R}^*)$ en $\text{spec}(A + BF)$ specificeerbaar, zie III.C. Merk op dat voor het specificeerbaar zijn van de polen van $A + BF$ nodig is dat (A, B) regelbaar is.

opmerking.

Met betrekking tot het probleem DDEPPP dient opgemerkt te worden dat het gewenste spectrum moet voldoen aan de eis:

$$\exists z \in \text{spectrum met } \text{im } z = 0. \quad (6)$$

Deze eis is een voldoende voorwaarde voor het kunnen splitsen van het gewenste spectrum in twee deelspectra. Zie ook de opmerking in III.C.

IV.B. DDEP: Disturbance Estimation Problem.

Uitgaande van systeem (*) wordt een estimator gezocht van de vorm:

$$\begin{aligned} \dot{\hat{w}} &= F \hat{w} + E y + R u \\ \hat{z} &= M \hat{w} \end{aligned} \quad (1)$$

zodanig dat \hat{z} een schatting is van de te schatten variabele z en deze schatting zodanig is dat de fout daarvan ($e = z - \hat{z}$) onafhankelijk is van de sturing u en de storing d .

We hebben dus:

(DDEP) bestaat een estimator (F, E, R, M) zodanig dat $e = z - \hat{z}$ onafhankelijk is van d en u . ?

(DDEPPP) bestaat een estimator zoals in (DDEP) waarbij bovendien het errorspectrum willekeurig te kiezen is. ?

Beide problemen zijn opgelost in /4/, waar nodige en voldoende voorwaarden zijn afgeleid, nl.:

$$(\text{DDEP}) \text{ is oplosbaar desda } \underline{S}^* \text{im}(G) \subset \ker H \quad (2)$$

$$(\text{DDEPPP}) \text{ is oplosbaar desda } \underline{N}^* \text{im}(G) \subset \ker H \quad (3)$$

Een oplossing voor (DDEP) is:

$$F = (A + L C) \pmod{\underline{S}}$$

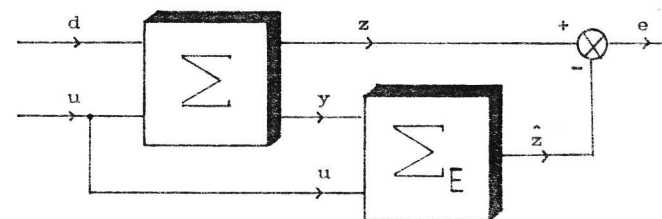
$$E = -L \pmod{\underline{S}}$$

$$R = B \pmod{\underline{S}}$$

$$M = H \pmod{\underline{S}}$$

waarbij $\underline{S} = \underline{S}^* \text{im}(G)$ en $L \in \underline{L}(\underline{S}^* \text{im } G)$.

Een oplossing voor (DDEPPP) heeft dezelfde gedaante, nu met $\underline{S} = \underline{N}^* \text{im}(G)$ en $L \in \underline{L}_p(\underline{N}^* \text{im } G)$.



bewijs:

noodzakelijkheid: zie /4/.

voldoende:

Aangetoond wordt, dat de voorwaarden voldoende zijn, wil de aangegeven oplossing inderdaad voldoen.

Pas weer een basistransformatie toe zodat $T^{-1}x = (s_1 \ s_2)^t$ met $s_1 \in \underline{S}^*$ en $s_2 \in \underline{X} \ \underline{S}^*$. We vinden dan:

$$\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u + \begin{bmatrix} G_1 \\ 0 \end{bmatrix} d$$

$$y = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}, \quad z = \begin{bmatrix} 0 & H_2 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = H_2 s_2$$

waarbij $G_2 = 0$ omdat $\text{im } G \subset \underline{S}^*$ (per definitie)

en $H_1 = 0$ omdat $\underline{S}^* \subset \ker H$ (de eis)

Voor DDEP vinden we nu uit (4) de volgende oplossing:

$$F = A_{22} + L_2 C_2$$

$$E = -L_2$$

$$R = B_2$$

$$M = H_2$$

waarbij L_2 uit: $L = T^{-1} \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} \in \underline{L}(\underline{S}^*)$.

Wegens de keuze van deze L is $A_{12} + L_2 C_1 = 0$.

Een korte berekening geeft dan:

$$\hat{z} = M \cdot w = H_2 \cdot w$$

$$z = H_2 \cdot s_2$$

zodat:

$$e = z - \hat{z} = H_2 \cdot (s_2 - w)$$

terwijl:

$$\dot{s}_2 - \dot{w} = (A_{22} + L_2 \cdot C_2) (s_2 - w)$$

zodat de fout e in de schatting inderdaad onafhankelijk is van de storing d .

Het bewijs van (3) gaat analoog. Merk op dat een $L \in \underline{L}_p(\underline{N}^*)$ juist zo is dat $\text{spec}(A_{22} + L_2 C_2)$ gelijk is aan de gewenste polen. Voor een constructie van zo'n L zie III.D.

IV.C. DDPM: Disturbance Decoupling problem with measurements

We gaan weer uit van systeem (*) en zoeken nu een feedback-processor van de vorm:

$$\begin{aligned} \dot{w} &= K w + L y \\ u &= M w + N y \end{aligned} \quad (1)$$

zodanig, dat de storingen d geen invloed hebben op de te regelen variabele z . (Merk op dat DDP in feite een speciaal geval hiervan is, waarbij $C = I$ in systeem (*). In dat geval kan voldaan worden met een directe terugkoppeling $u = F y = F x$ in plaats van (1)).

Wiskundig gezien komt de eis weer neer op het nul zijn van de transfermatrix van d naar z van het closedloop systeem.

Dat closedloop systeem heeft hier de vorm:

$$\begin{bmatrix} \dot{x} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} A + B N C & B M \\ L C & K \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} + \begin{bmatrix} G \\ 0 \end{bmatrix} d$$

$$z = \begin{bmatrix} H & 0 \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} \quad (2)$$

We hebben nu:

(DDPM) bestaat een feedbackprocessor (K, L, M, N) zodanig dat de transfermatrix van (2) nul is. ?

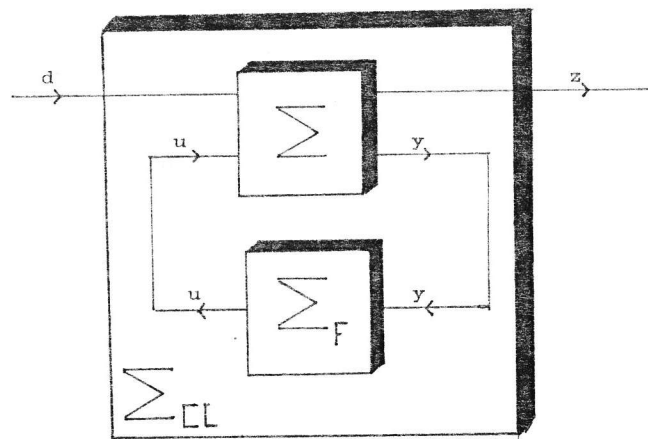
(DDPMPP) bestaat een feedbackprocessor als in (DDPM) waarbij de polen van het closedloop systeem willekeurig zijn te kiezen. ?

Voor beide problemen zijn nodige en voldoende voorwaarden afgeleid in /4/. Behalve in /4/ wordt op dit probleem ook ingegaan in /5/. Zowel Willems in /4/ als Schumacher in /5/ dragen ook oplossingen aan. De oplossing van Willems is wellicht inzichtelijker maar die van Schumacher leidt in het algemeen tot lagere dimensies. Schumacher's feedbackprocessor is altijd van dezelfde dimensie als het systeem zelf, terwijl bij Willems een en ander kan oplopen tot drie keer die dimensie.

Het volgende is afgeleid:

$$\text{(DDPM) is oplosbaar desda } \underline{S}^* \text{im}(G) \subset \underline{V}^* \ker(H) \quad (3)$$

$$\begin{aligned} \text{(DDPMPP) is oplosbaar desda } \underline{N}^* \text{im}(G) \subset \underline{R}^* \ker(H) \\ \text{en } (A, B) \text{ regelbaar} \\ \text{en } (A, C) \text{ waarneembaar} \end{aligned} \quad (4)$$



Schumacher draagt nu de volgende oplossing aan:

$$\begin{aligned} K &= A + B F_v + L_s C - B K_{vs} C \\ L &= -L_s + B K_{vs} \\ M &= F_v - K_{vs} C \\ N &= K_{vs} \end{aligned} \quad (5)$$

Hierbij is:

$$\begin{aligned} F_v &\in \mathcal{F}(\mathcal{V}^{\text{ker}}(H)) \\ L_s &\in \mathcal{L}(\mathcal{S}^{\text{im}}(G)) \\ K_{vs} &\in \mathcal{K}(\mathcal{V}^{\text{ker}}(H), \mathcal{S}^{\text{im}}(G)) \end{aligned}$$

ingeval we (DDPM) oplossen en:

$$\begin{aligned} F_v &\in \mathcal{F}_p(\mathcal{R}^{\text{ker}}(H)) \text{ met spec } (A + B F) \text{ specificieerbaar.} \\ L_s &\in \mathcal{L}_p(\mathcal{N}^{\text{im}}(G)) \text{ met spec } (A + L C) \text{ specificieerbaar.} \\ K_{vs} &\in \mathcal{K}(\mathcal{R}^{\text{ker}}(H), \mathcal{N}^{\text{im}}(G)) \end{aligned}$$

ingeval we (DDPMPP) oplossen.

bewijs:

noodzakelijkheid: zie /4/

voldoende: We bewijzen dat de voorwaarde $\underline{S} \subset \underline{V}$ voldoende is opdat de processor, zoals gedefinieerd in (5) inderdaad voldoet aan de gestelde eisen, d.w.z. de transferfunctie van het closedloop-systeem nul. Het bewijs is van Willems*.

We definiëren de volgende deelruimte:

$$\mathcal{L} = \left\{ \begin{bmatrix} x \\ w \end{bmatrix} \text{ met } \begin{bmatrix} x \\ w \end{bmatrix} = \begin{bmatrix} p \\ o \end{bmatrix} + \begin{bmatrix} q \\ q \end{bmatrix}, p \in \underline{S} \text{ en } q \in \underline{V} \right\}$$

* We gebruiken de afkorting \underline{V} voor \mathcal{V}^{ker} resp \mathcal{R}^{ker} en \underline{S} voor \mathcal{S}^{im} resp. \mathcal{N}^{im} al na gelang we (DDPM) resp. (DDPMPP) bewijzen.

$$\text{Laat verder } A_{cl} = \begin{bmatrix} A + BNC & BM \\ LC & K \end{bmatrix} \quad (8)$$

dan volgt dat \mathcal{L} A_{cl} -invariant is:

Uit (6) volgt dat

$$A_{cl} = \begin{bmatrix} A + B K_{vs} C & B F_v - B K_{vs} C \\ -L_s C + B K_{vs} C & A + B F_v + L_s C - B K_{vs} C \end{bmatrix} \quad (9)$$

en hieruit volgt:

$$\begin{aligned} A_{cl} \begin{bmatrix} p + q \\ q \end{bmatrix} &= \begin{bmatrix} (A + B K_{vs} C) p + (A + B F_v) q \\ (A + B K_{vs} C) p + (A + B F_v) q - (A + L_s C) p \end{bmatrix} \\ &= \begin{bmatrix} q' + q'' \\ q' + q'' - p' \end{bmatrix} \end{aligned} \quad (10)$$

met $q' \in \underline{V}$, $q'' \in \underline{V}$ en $p' \in \underline{S}$.

De laatste stap volgt uit de definities van F_v , L_s en K_{vs} :

$$(A + B F_v) \underline{V} \subset \underline{V}, (A + L_s C) \underline{S} \subset \underline{S} \text{ en } (A + B K_{vs} C) \underline{S} \subset \underline{V}.$$

Uit de eis voor oplosbaarheid: $\underline{S} \subset \underline{V}$ volgt het bestaan van een $v \in \underline{V}$ zodat met $v' = q' + q'' \in \underline{V}$ en p' alsboven geldt:

$$v' = v + p'$$

(10) gaat dan over in: $\begin{bmatrix} v' - p' \\ v \end{bmatrix}$ hetgeen een element is van \mathcal{L} .

Hiermee is bewezen:

$$A_{cl} \mathcal{L} \subset \mathcal{L} \quad (11)$$

Verder geldt:

$$\text{im} \begin{bmatrix} G \\ 0 \end{bmatrix} \subset \mathcal{L} \quad (12)$$

(uit $\text{im } G \subset \underline{S}$)

$$\mathcal{L} \subset \ker[H \ 0] \quad (13)$$

(uit $\underline{V} \subset \ker H$)

De laatste drie gegevens geven aanleiding tot

$$A_{cl} \cdot \text{im} \begin{bmatrix} G \\ 0 \end{bmatrix} \subset A_{cl} \mathcal{L} \subset \mathcal{L} \subset \ker[H \ 0] \quad (14)$$

waarna uit (14) volgt

$$(I_s - A_{cl})^{-1} \text{im} \begin{bmatrix} G \\ 0 \end{bmatrix} \subset \ker[H \ 0]$$

$$\text{en inderdaad de transferfunctie } [H \ 0] (I_s - A_{cl})^{-1} \begin{bmatrix} G \\ 0 \end{bmatrix} = 0 \quad (15)$$

Hiermee is (DDPM) volledig bewezen.

Voor (DDPMPP) passen we de transformatie $S = \begin{bmatrix} I & I \\ 0 & I \end{bmatrix}$ toe.

Dit impliceert:

$$S^{-1} \cdot A_{cl} \cdot S = \begin{bmatrix} A + L_s C & 0 \\ -L_s C + B K_{vs} C & A + B F_v \end{bmatrix} \quad (16)$$

waaruit:

$$\text{spec}(A_{cl}) = \text{spec}(A + L_s C) \cup \text{spec}(A + B F_v) \quad (17)$$

opmerking:

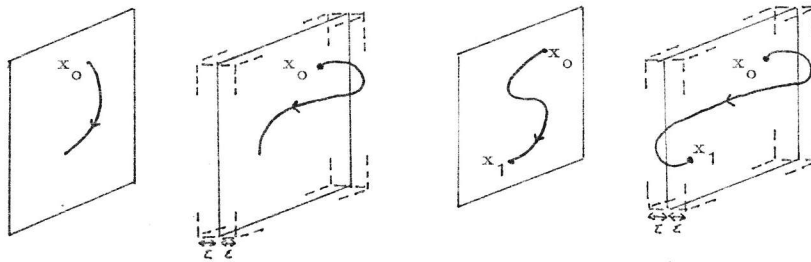
Ook hier dient een opmerking gemaakt te worden over het gewenste spectrum. In de eerste plaats moet een opdeling als in (17) mogelijk zijn. Voorts moeten de beide zo ontstane deelspectra nog weer eens opgedeeld kunnen worden.

Een voldoende voorwaarde is de eis dat beide deelspectra in (17) elk een zuiver reëel element bevatten. Deze eis is echter niet noodzakelijk. Zie ook de opmerkingen bij III.C. en III.D.

In LIDIA worden achtereenvolgens de polen gevraagd behorend bij R^* , bij $X \setminus R^*$, bij N^* en bij $X \setminus N^*$.

hoofdstuk V. BIJNA STORINGSONTKOPPELING

Behalve de invariante deelruimten zoals gedefinieerd in hoofdstuk III bestaan hiervan ook zg. 'bijna invariante' versies. Deze worden o.a. in /12/ en /13/ door Willems geïntroduceerd. In dit hoofdstuk wordt volstaan met het geven van definities en methoden om die deelruimten te kunnen berekenen. Voor bewijzen en toepassingen van deze deelruimten op de 'almost'-versies van de storingsontkoppelingsproblemen wordt verwezen naar /12/ en /13/.



Het 'bijna'-idee komt ondermeer tot uiting met bovenstaande figuren waarin vlnr. zijn uitgebeeld een (A,B) invariante deelruimte, een bijna (A,B) invariante deelruimte, een regelbaarheidsdeelruimte en een bijna regelbaarheidsdeelruimte.

V.A. Definities.Bijna (A,B) invariante deelruimte

Uitgaande van het systeem $\dot{x} = A x + B u$ heet een deelruimte $V_a \subset X$ Almost controlled invariant of bijna (A,B)-invariant als:

$\forall x_0 \in V_a \exists \varepsilon > 0 \exists$ trajectory x zodanig dat $x(0) = x_0$ en $d(x(t), V_a) \leq \varepsilon$ voor $\forall t$.

Bijna regelbaarheidsdeelruimte

Uitgaande van het systeem $\dot{x} = A x + B u$ heet een deelruimte $R_a \subset X$ Almost controllability of bijna regelbaarheidsdeelruimte als:

$\forall x_0, x_1 \in R_a \exists T > 0$ zodanig dat $\forall \varepsilon > 0 \exists$ trajectory x met $x(0) = x_0$, $x(T) = x_1$ en $d(x(t), R_a) \leq \varepsilon$ voor $\forall t$.

grootste element.

Evenals in het niet-bijna-geval geldt ook hier dat de verzameling van ruimtes V_a resp. R_a gesloten is onder optelling en dientengevolge bestaat van elk een grootste element, aangeduid met resp.

V_a^* en R_a^* . We hebben nu:

$$\begin{aligned} V_a^*(K) &= \sup \left\{ V_a \mid V_a \text{ is bijna (A,B)-invariant en } V_a \subset K \right\} \\ R_a^*(K) &= \sup \left\{ R_a \mid R_a \text{ is bijna regelbaarheidsdeelruimte en } R_a \subset K \right\} \end{aligned} \quad (1), (2)$$

Als bijzonderheid (zie th. 2 pag. 237 in /12/) geldt:

$$V_a^*(K) = V_a^*(K) + R_a^*(K) \quad (3)$$

V_a^* en R_a^* zijn eveneens te bepalen door toedoen van de algoritmes (ISA) en (ACSA). Geef met V_K^ω de oplossing aan van (ISA) en met R_K^ω die van (ACSA) (Zie ook hoofdstuk II en III). Dan is:

$$V_a^*(K) = V_K^\omega \quad (4)$$

$$R_a^*(K) = R_K^\omega \quad (5)$$

$$V_a^*(K) = V_K^\omega + R_K^\omega \quad (6)$$

$$R_a^*(K) = V_K^\omega \cap R_K^\omega = V_{R_K^\omega}^\omega = R_{V_K^\omega}^\omega \quad (7)$$

(zie /12/).

(4) en (7) waren reeds bekend (zie hoofdstuk II). (5) en (6) geven ons de mogelijkheid tot het berekenen van $V_a^*(K)$ en $R_a^*(K)$. Uit (5) zal het ogenblikkelijk duidelijk zijn waarom (ACSA) niet slechts Controllability Subspace Algorithm (CSA) wordt genoemd zoals in /2/.

Bijna (A,C)-invariante deelruimte

Uitgaande van $\dot{x} = A x$ en $y = C x$ heet een deelruimte $\underline{S}_a \subset \underline{X}$ almost conditional invariant of bijna (A,C) invariant als $\forall x_0 \in \underline{S}_a$ en $\exists \xi > 0$ \exists matrices K en L zodanig dat voor $w(t)$ met $\dot{w} = K w + L y$ en $w(0) = 0$ geldt:

$$\|w(t) - x(t) \pmod{\underline{S}_a}\| \leq \xi, \forall t.$$
Bijna niet-waarneembaarheidsdeelruimte

Wederom met $\dot{x} = A x$ en $y = C x$ heet een deelruimte $\underline{N}_a \subset \underline{X}$ almost complementary observability of bijna-niet-waarneembaarheidsdeelruimte als $\forall x_0 \in \underline{N}_a$ en polynomen p met graad p = toestandsdimensie - $\dim \underline{N}_a$ en $\forall \xi > 0$, \exists matrices K en L, zodanig dat voor $e := w - x \pmod{\underline{N}_a}$ de volgende uitdrukking geldt:

$$e(t) = e^{Kt} e(0) + d(t), \text{ waarbij:}$$

$$\chi_K = p \text{ en } \|d(t)\| \leq \xi, \forall t.$$

Evenals in het niet-bijna-geval duidt spec (K) hier het errorspectrum aan.

kleinste element

De verzameling van ruimtes \underline{S}_a resp. \underline{N}_a bezitten een kleinste element omdat ze gesloten zijn onder intersectie. We vinden:

$$\begin{aligned} \underline{S}_a^*(K) &= \inf \{ \underline{S}_a \mid \underline{S}_a \text{ is bijna (A,C) invariant en } K \subset \underline{S}_a \} \\ \underline{N}_a^*(K) &= \inf \{ \underline{N}_a \mid \underline{N}_a \text{ is bijna niet-waarneembaarheidsdeelruimte en } K \subset \underline{N}_a \} \end{aligned} \quad (8), (9)$$

In /13/ worden de volgende dualiteitseigenschappen afgeleid:

$$\underline{S}^* \text{ im } G = (\underline{V}^* \ker G^t)^\perp \quad (10)$$

$$\underline{N}^* \text{ im } G = (\underline{R}^* \ker G^t)^\perp \quad (11)$$

$$\underline{S}_a^* \text{ im } G = (\underline{V}_a^* \ker G^t)^\perp \quad (12)$$

$$\underline{N}_a^* \text{ im } G = (\underline{R}_a^* \ker G^t)^\perp \quad (13)$$

Geef met \underline{V}_K^ω de oplossing aan van (ISA), met \underline{R}_K^ω de oplossing van (ACSA) en evenzo \underline{S}_K^ω en \underline{N}_K^ω voor de oplossingen van de duale algoritmen (zie /13/) dan is af te leiden:

$$\underline{S}^* \text{ im } G = \underline{S}_{\text{im } G}^\omega = (\underline{V}_{\ker G^t}^\omega)^\perp \quad (14)$$

$$\underline{N}_a^* \text{ im } G = \underline{N}_{\text{im } G}^\omega = (\underline{R}_{\ker G^t}^\omega)^\perp \quad (15)$$

$$\underline{S}_a^* \text{ im } G = \underline{S}_{\text{im } G}^\omega \cap \underline{N}_{\text{im } G}^\omega \quad (16)$$

$$\begin{aligned} \underline{N}^* \text{ im } G &= \underline{S}_{\text{im } G}^\omega + \underline{N}_{\text{im } G}^\omega = \underline{S}_{\underline{N}_{\text{im } G}^\omega}^\omega = \underline{N}_{\underline{S}_{\text{im } G}^\omega}^\omega \\ &= (\underline{R}_{(\underline{S}_{\text{im } G}^\omega)^\perp}^\omega)^\perp = (\underline{R}_{\underline{V}_{\ker G^t}^\omega}^\omega)^\perp \end{aligned} \quad (17)$$

Bovenstaande vergelijkingen volgen m.b.v. de dualiteitseigenschappen direct uit (4) - (7).

Ook nu blijkt dat we aan de algoritmen (ISA) en (ACSA) voldeende hebben om alle deelruimten te berekenen.

V.B. Implementaties.

Voor het bepalen van $\underline{V}^*_{\text{ker } H}$, $\underline{R}^*_{\text{ker } H}$, $\underline{S}^*_{\text{im } G}$ en $\underline{N}^*_{\text{im } G}$ zie de hoofdstukken II en III.

 $\underline{R}^*_{\text{ker } H}$

Voor het bepalen van de grootste bijna-regelbaarheidsdeelruimte in $\text{ker } H$ passen we (ACSA) toe met $\underline{K} = \text{ker } H$.

 $\underline{V}^*_{\text{ker } H}$

Voor het bepalen van de grootste bijna-(A,B)-invariante deelruimte in $\text{ker } H$ passen we V.A.(6) toe.

Bepaal een matrix T , zodanig dat:

$$T^* \cdot \begin{bmatrix} V & R \end{bmatrix} = \begin{bmatrix} S \\ 0 \end{bmatrix}$$

waarbij $\text{im } V = \underline{V}^*_{\text{ker } H}$, $\text{im } R = \underline{R}^*_{\text{ker } H}$ en S volle rijrang heeft. Als rr de rijrang is van S dan spannen de eerste rr kolommen van de matrix T juist de ruimte $\text{im } \begin{bmatrix} V & R \end{bmatrix}$ op, en dat is, volgens V.A.(6) juist $\underline{V}^*_{\text{ker } H}$.

 $\underline{N}^*_{\text{im } G}$

Voor het bepalen van de kleinste bijna-niet-waarneembaarheidsdeelruimte die $\text{im } G$ omvat passen we (ACSA) toe op het duale systeem en nemen $\underline{K} = \text{ker } G^t$.

 $\underline{S}^*_{\text{im } G}$

Voor het bepalen van de kleinste bijna (A,C)-invariante deelruimte die $\text{im } G$ omvat passen we V.A.(16) toe.

Met $\text{im } S = \underline{S}^*_{\text{im } G}$ en $\text{im } N = \underline{N}^*_{\text{im } G}$ volgt uit V.A.(16):

$$\underline{S}^*_{\text{im } G} = \text{im } S \wedge \text{im } N = ((\text{im } S)^\perp + (\text{im } N)^\perp)^\perp$$

Aangezien in LIDIA steeds de gehele basistransformatie T_s resp. T_n wordt bewaard waarbij de eerste p kolommen van T_s de ruimte $\underline{S}^*_{\text{im } G}$ opspannen en de eerste q kolommen van T_n de ruimte $\underline{N}^*_{\text{im } G}$ (met $p = \dim \underline{S}^*_{\text{im } G}$ en $q = \dim \underline{N}^*_{\text{im } G}$) vinden we $(\text{im } S)^\perp$ resp. $(\text{im } N)^\perp$ als het opspansel van de laatste $n - p$ kolommen van T_s , resp. de laatste $n - q$ kolommen van T_n . Veronderstel daartoe:

$$\text{im } S^\perp = \text{sp} (t_{s_{p+1}}, \dots, t_{s_n}) \text{ en } \text{im } N^\perp = \text{sp} (t_{n_{q+1}}, \dots, t_{n_n}).$$

Bepaal nu een matrix T zodanig dat:

$$T^* \cdot \begin{bmatrix} S^\perp & N^\perp \end{bmatrix} = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

met R volle rijrang rr . Dan spannen de laatste $rr+1$ tem. n -de kolommen van T juist $\text{im } \begin{bmatrix} S^\perp & N^\perp \end{bmatrix}$ op en dus $\underline{S}^*_{\text{im } G}$.

hoofdstuk VI. VOORBEELDEN

Aangezien de hiervoor beschreven storingsontkoppelingstheorie pas echt interessant wordt als er ook werkelijk toepassingen voor te vinden zijn is een hoofdstuk met enkele voorbeelden c.q. toepassingen toegevoegd.

VI.A. Destillatiekolom.

veronderstellingen

T.a.v. de destillatiekolom zijn de volgende veronderstellingen genomen:

- de vluchtigheid in de gehele kolom is constant
- iedere schotel is perfect, d.w.z. is 100 % efficiënt
- er is een toevoerschotel waaraan een enkele voedingsstroom in de vorm van verzadigde vloeistof (op kookpunt) wordt toegevoerd.
- de gas-overhead in de bovenste schotel condenseert volledig in de condenser.
- de achterblijvende vloeistof op iedere schotel is constant en perfect gemixed
- het achterblijvende gas is verwaarloosbaar
- de vloeistof-flow door de schotels is constant, de gas-flow eveneens.

vergelijkingen

Uit de constante vluchtigheid volgt de volgende vergelijking die het gas-vloeistof-evenwicht aangeeft:

$$y_j = \frac{ax_j}{1 + (a-1)x_j} \quad (1)$$

waarbij: x_j : vloeistof-samenstelling op j-de schotel

y_j : gas-samenstelling j-de schotel

a : relatieve vluchtigheid.

In de condenser geldt:

$$\dot{x}_D = \frac{v_n}{H_D} (y_1 - x_D) \quad (2)$$

Voor de verrijkings-schotels: $(1 \leq j \leq f-1)$

$$\dot{x}_j = \frac{v_n}{H} (y_{j+1} - y_j) + \frac{L_n}{H} (x_{j-1} - x_j) \quad (3)$$

Voor de voedingsschotel:

$$\dot{x}_f = \frac{F}{H} (x_F - x_f) + \frac{v_n}{H} (y_{f+1} - y_f) + \frac{L_n}{H} (x_{f-1} - x_f) \quad (4)$$

Voor de verarmingschotels: $(f+1 \leq j \leq N)$

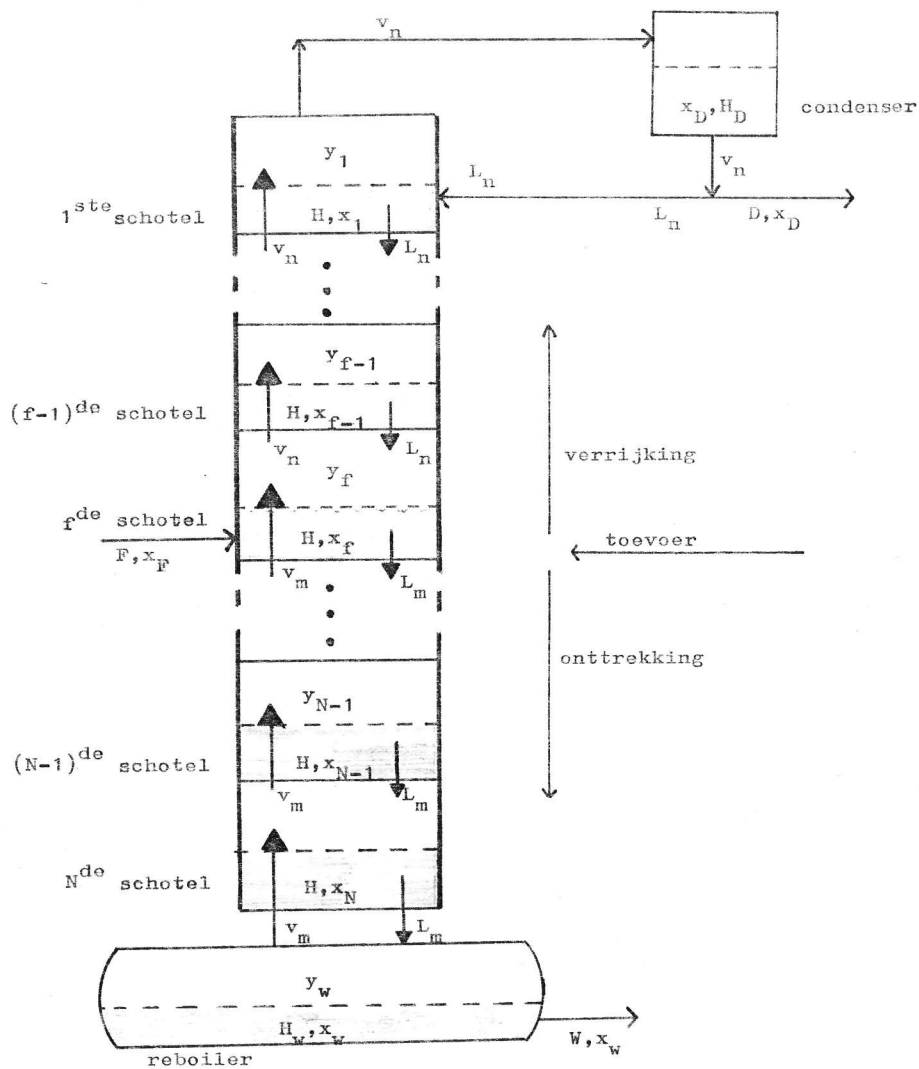
$$\dot{x}_j = \frac{v_m}{H} (y_{j+1} - y_j) + \frac{L_m}{H} (x_{j-1} - x_j) \quad (5)$$

En voor de reboiler:

$$\dot{x}_W = \frac{v_m}{H_W} (x_W - y_W) + \frac{L_m}{H_W} (x_N - x_W) \quad (6)$$

Bij dit alles is:

$$\begin{aligned} L_m &= L_n + F \\ V_n &= v_m \end{aligned} \quad (7)$$



Schematische voorstelling van de destillatiekolom.

Waarbij: D : flow-rate van het destillaat

W : idem van het bodem-produkt

V : idem van de gasstroom door de schotel

L_n : idem van de vloeistofstroom door de schotels in het verrijkingsgedeelte

L_m : idem van de vloeistofstroom door de schotels in het verarmingsgedeelte

H : achterblijvende vloeistof op iedere schotel

H_W : idem in de reboiler (bodem)

H_D : idem in de condenser

F : flow-rate van de toevoerstream (constant)

De flowrate wordt uitgedrukt in mol/minuut, H, H_W en H_D worden uitgedrukt in mol.

Er blijkt een evenwichtstoestand te bestaan. Rond deze toestand wordt gelineariseerd. Het resultaat is een systeem van de vorm:

$$\dot{\bar{x}} = A\bar{x} + B\bar{u} + G\bar{d} \quad (8)$$

$$\bar{z} = H\bar{x}$$

met: $\bar{x} = (\Delta x_D, \Delta x_1, \dots, \Delta x_N, \Delta x_W)$ de toestand

$\bar{u} = (\Delta L_n, \Delta L_m)$ de sturing

$\bar{d} = (\Delta F, \Delta x_F)$ de verstoring

$\bar{z} = (\Delta x_D, \Delta x_W)$ de te regelen toestand

waarbij Δ betekent: afwijking t.o.v. evenwicht.

In /14/ zijn simulaties uitgevoerd waaruit bleek, dat het gelineariseerde systeem (8) zich net zo gedraagt als het niet-lineaire systeem (1)-(7), mits de afwijkingen t.o.v. het evenwichtspunt niet te groot werden. Dit is de reden dat hier slechts simulaties zijn uitgevoerd met het gelineariseerde systeem.

In /14/ zijn de matrices A, B, C, G en H numeriek bepaald:

A is tridiagonaal, met:

$$A(i,i) = (-0.174, -0.943, -0.991, -1.051, -1.118, -1.584, -1.640, -1.721, -1.823, -1.943, -0.171) \quad i=1, \dots, 11$$

$$A(i+1,i) = (0.522, 0.522, 0.522, 0.522, 0.522, 0.922, 0.922, 0.922, 0.922, 0.115) \quad i=1, \dots, 10$$

$$A(i,i+1) = (0.105, 0.469, 0.529, 0.596, 0.596, 0.718, 0.799, 0.901, 1.021, 1.142) \quad i=1, \dots, 10$$

De andere elementen van de matrix A zijn nul.

$$B = \begin{bmatrix} 0 & 0 \\ 0.00328 & -0.00244 \\ 0.00384 & -0.00288 \\ 0.00400 & -0.00304 \\ 0.00376 & -0.00280 \\ 0.00308 & -0.00232 \\ 0.00236 & -0.00312 \\ 0.00288 & -0.00382 \\ 0.00308 & -0.00412 \\ 0.00300 & -0.00396 \\ 0.00032 & -0.00042 \end{bmatrix}, \quad G = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0.4 \\ 0.00236 & 0 \\ 0.00288 & 0 \\ 0.00308 & 0 \\ 0.00300 & 0 \\ 0.00032 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Lidia berekent bij bovenstaande matrices de volgende deelurimten:

$$V^* \text{ker } H = \text{sp} (e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10})$$

$$R^* \text{ker } H = 0$$

$$S^* \text{im } G = \text{sp} \begin{bmatrix} -0.023 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.619 & -0.072 & -0.107 & 0 & 0 & 0 & 0 & 0 \\ -0.495 & -0.433 & -0.432 & 0.209 & 0 & 0 & 0 & 0 \\ 0.446 & -0.544 & -0.042 & 0.412 & -0.371 & 0 & 0 & 0 \\ -0.729 & -0.348 & 0.063 & -0.088 & -0.230 & 0.543 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0.047 & 0.225 & -0.407 & 0.057 & 0.148 & 0.840 & 0 & 0 \\ -0.159 & 0.355 & -0.006 & -0.140 & -0.888 & 0 & 0 & 0 \\ -0.129 & 0.368 & 0.131 & 0.870 & 0 & 0 & 0 & 0 \\ 0.272 & 0.264 & -0.881 & 0 & 0 & 0 & 0 & 0 \\ -0.220 & 0.091 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$N^* \text{im } G = \mathbb{R}^{11}$$

en concludeert dat geen van de problemen DDP, DDEP en DDPM oplosbaar is.

Gebruiken we in de berekeningen slechts de tweede kolom van G (d.w.z. we beperken ons tot de tweede storingscomponent), dan vinden we dezelfde $V^* \text{ker } H$, $R^* \text{ker } H$ en $S^* \text{im } G$ en is $N^* \text{im } G = S^* \text{im } G$.

Nu blijkt DDP oplosbaar te zijn en een oplossing is:

$$F = \begin{bmatrix} 0 & 0 & -330.06 & 0 & 0 & 0 & 0 & 0 & 0 & 470.17 & 0 \\ 0 & 0 & -251.47 & 0 & 0 & 0 & 0 & 0 & 0 & 632.04 & 0 \end{bmatrix}$$

De problemen DDEP en DDPM zijn nog steeds niet oplosbaar.

We proberen, of een extra waarneming kan helpen. Daartoe nemen

$$\text{we } y_3 = x_6.$$

Met A, B en H als voren en:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0.4 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^t$$

vinden we:

$$V^* \text{ker } H = \text{sp} (e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10})$$

$$R^* \text{ker } H = 0$$

$$S^* \text{im } G = \text{sp} (e_6)$$

$$N^* \text{im } G = \text{sp} (e_6)$$

en kunnen we concluderen, dat zowel DDEP als DDPM nu oplosbaar zijn.

Intuitief lijken de waarnemingen y_1 en y_2 weinig bij te dragen aan de regeling en inderdaad, nemen we alleen y_3 als waarneming, dan vinden we:

$$A, B, H \text{ en } G \text{ als voren en } C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

$$V^* \text{ker } H = \text{sp} (e_3, e_4, e_6, e_5, e_7, e_8, e_9, e_{10})$$

$$R^* \text{ker } H = 0$$

$$S^* \text{im } G = \text{sp} (e_6)$$

$$N^* \text{im } G = \mathbb{R}^{11}.$$

Opnieuw zijn DDEP en DDPM oplosbaar.

Omdat (A,B) niet regelbaar is, zijn DDPPP en DDPMPP nooit oplosbaar.

conclusies.

Het blijkt onmogelijk, de beide storings, resp. verstoring in de samenstelling van de toevoerstroom en verstoring in de grootte van die toevoerstroom, te ontkoppelen van het eindresultaat, de vloeistofsamenstelling in de condensor en de reboiler.

Indien we de vloeistofsamenstelling van elke schotel kunnen waarnemen, blijkt de verstoring in de samenstelling van de toevoerstroom te kunnen worden ontkoppeld van het eindresultaat (DDP).

Deze ont koppeling is niet mogelijk als we alleen de eindresultaten waarnemen (hetgeen logisch is, als men bedenkt dat de invloed van de storing dan pas waargenomen wordt op het moment dat deze eigenlijk al weggeregeld had moeten zijn.).

Toepassing van alleen de waarneming van de samenstelling van de vloeistof op de toevoerschotel geeft de mogelijkheid deze ont koppeling wel uit te voeren (DDPM). Dit kan een indicatie zijn, dat de volgorde:

verstoring - waarneming - regeling - eindresultaat van belang is. In dit geval wordt eerst een verstoring 'veroorzaakt' en daarna in de vloeistofsamenstelling op de voedingsschotel waargenomen. Door vervolgens de vloeistofdoorstroming te regelen is het mogelijk de verstoring 'weg te maken' voordat deze het eindresultaat kan beïnvloeden.

VI.B. Koersregeling bij zeeschepen.

Veronderstellingen

T.a.v. het te modelleren zeeschip zijn o.a. de volgende veronderstellingen genomen:

- het schip vaart met een constante snelheid rechtvooruit (in de x-richting).
- het zwaartepunt van het schip bevindt zich in het x-z-vlak
- lading en verdeling van lading blijft constant

vergelijkingen

De wet van behoud van impuls en de wet van behoud van impulsmoment geven:

$$m \left(\frac{dv}{dt} + r u_0 + x_G \frac{dr}{dt} \right) = Y(v, r, \delta, \dot{v}, \dot{r}) \quad (1)$$

$$I_z \frac{dr}{dt} + m x_G \left(\frac{dv}{dt} + r u_0 \right) = N(v, r, \delta, \dot{v}, \dot{r}) \quad (2)$$

Hierbij is:

- u_0 - (constante) snelheid van het schip in de x-richting
- v - snelheid in de y - richting (slingersnelheid, sway-vel.)
- r - hoeksnelheid rond z-as
- δ - stand van het roer
- ψ - richting van het schip
- m - massa van het schip (constant)
- I_z - traagheidsmoment rond de z-as
- x_G - x-coördinaat van het zwaartepunt
- Y - component van de hydrodynamische kracht in y-richting
- N - component van het koppel, door de hydrodynamische kracht veroorzaakt, in de z-richting

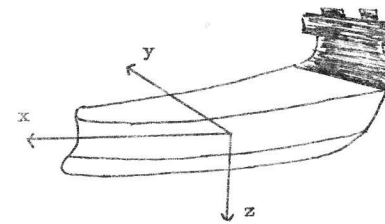
Als verstoring wordt beschouwd de wind. We gaan ervanuit dat deze wind slechts verstoringen teweeg brengt in de vorm van een additieve kracht Y_w en een addatief draaimoment N_w . Deze zijn afhankelijk van de richting van het schip, ψ , en tevens van de windrichting, θ , de richting van de relatieve windvector w_{rel} , zie tekening, zoals die op het schip wordt waargenomen.

We krijgen dan de vergelijkingen:

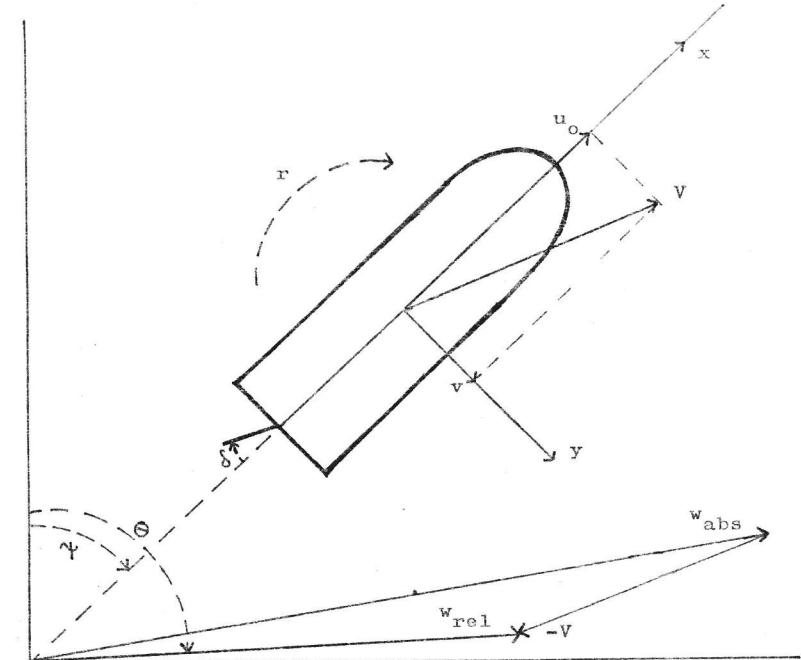
$$m \left(\frac{dv}{dt} + r u_0 + x_G \frac{dr}{dt} \right) = Y + Y(\psi, \theta) \quad (3)$$

$$I_z \frac{dr}{dt} + m x_G \left(\frac{dv}{dt} + r u_0 \right) = N + N(\psi, \theta) \quad (4)$$

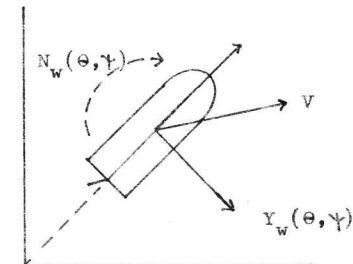
Er bestaat een evenwichtspunt. Als $\dot{v} = \dot{r} = 0$ voldoet in ieder geval $v = r = Y = N = 0$, uitgaande van de veronderstelling dat het schip symmetrisch is rond het x-z-vlak. Linearisatie rond dit evenwicht geeft het volgende lineaire systeem:



Assenstelsel



Model



Verstoringen

$$\frac{d}{dt} \begin{bmatrix} v \\ r \\ \psi \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ r \\ \psi \end{bmatrix} + \begin{bmatrix} b_{11} \\ b_{21} \\ 0 \end{bmatrix} \delta + \begin{bmatrix} e_1 \\ e_2 \\ 0 \end{bmatrix}$$

$$z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ r \\ \psi \end{bmatrix} \quad y = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ r \\ \psi \end{bmatrix}$$

De waarden a_{ij} en b_{ij} zijn afhankelijk van de schip-parameters. De waarden a_{13} en a_{23} zijn afhankelijk van de parameters van de constante factor in de windverstoring. Zuidweg geeft in /16/ formules voor a_{13} en a_{23} waardoor bij opgegeven windsnelheid en -richting deze factoren kunnen worden berekend. De overige waarden van a_{ij} en b_{ij} zijn te vinden in /15/ voor een vijftal verschillende schepen.

De factoren e_1 en e_2 zijn de fluctuaties in windsnelheid en -richting en worden hier als de storing opgevat.

Ten overvloede zij hierbij vermeld, dat wordt uitgegaan van een 'verstoring', uitsluitend het gevolg van de wind, waarbij deze is opgedeeld in een constant deel (de gemiddelde snelheid en richting) en een fluctuerend deel (de afwijking in snelheid en richting). Het constante deel wordt in de systeembeschrijving opgenomen (de factoren a_{13} en a_{23}) en het fluctuerende deel wordt als storing beschouwd.

In LIDIA zijn voor alle vijf schepen, vermeld in /15/, de deelruimten berekend, waarbij de wind verondersteld werd te waaien met een gemiddelde snelheid van 12 m/s en in een hoek van 45° tov. van het schip. Voor de /15/ vermelde tanker (210.000 dwt) leverde dat het volgende op:

$$A = \begin{bmatrix} -0.466 & -0.196 & -0.0141 \\ -0.302 & -0.156 & 0.0060 \\ 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0.176 \\ -0.124 \\ 0 \end{bmatrix}, \quad G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

$$V^* \ker H = \text{sp}(e_1)$$

$$R^* \ker H = \emptyset$$

$$S^*_{\text{im}} G = \mathbb{R}^3$$

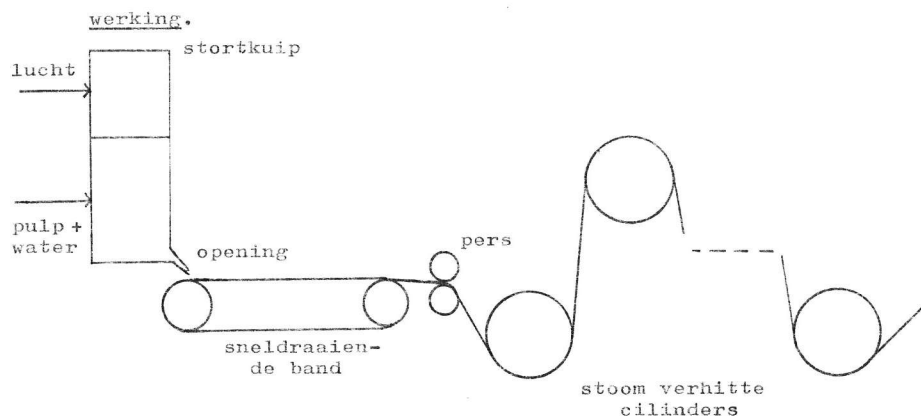
$$N^*_{\text{im}} G = \mathbb{R}^3$$

waaruit geconcludeerd kan worden, dat geen van de problemen DDP, DDEP en DDPM oplosbaar is. Wordt alleen de eerste kolom of de tweede kolom van G beschouwd dan vinden we dezelfde deelruimten. Bij beschouwing van de eerste kolom is DDP dan wel oplosbaar, maar dit resultaat heeft weinig praktische betekenis.

conclusies

Het probleem in de huidige vorm biedt geen (praktische) mogelijkheid tot storingsontkoppeling. Het is niet de aangewezen weg dit probleem met de ontkoppelings-theorie aan te pakken. In de literatuur zijn andere methoden met succes geprobeerd. Nemen we aan dat de factoren e_1 en e_2 voorgesteld kunnen worden door middel van witte ruis (en deze keuze blijkt inderdaad niet onredelijk te zijn) dan kunnen we de optimal-control-theorie op het probleem loslaten. Een en ander blijkt dan succesvol te werken. Zie /18/.

VI.C. Een papierfabriek.



In de zg. Pourdrinier papier-machine wordt de grondstof, de pulp, sterk verdund met water en deze gehele vezelbrij wordt in een stortkuip gepompt. Aan de onderkant hiervan is een opening waardoor de vezelbrij wordt uitgestort op een snel draaiend koper- of plastic doek. Hier vormt zich een papierbaan, die daarna d.m.v. persen en door stoom verhitte cilinders wordt geperst en gedroogd. De kwaliteit van het papier is in hoge mate afhankelijk van de relatieve snelheid waarmee de vezelbrij uit de stortkuip op de band terecht komt. Verantwoordelijk voor deze snelheid is de hoogte van de vezelbrij in de stortkuip. Verder wordt er lucht in de stortkuip gepompt om een bepaalde druk op de daar aanwezige brij te verkrijgen. De 'flow' van de brij door de opening hangt dus af van de hoogte van de brij in de stortkuip en de druk van de lucht erboven.

Een schematische voorstelling van het geheel is hierboven te vinden. Voorts is op de volgende pagina's een gedetailleerde tekening opgenomen waarin het complete proces van aanvoer van de pulp tot afvoer van het papier is weergegeven. Deze tekening komt van Van Gelder Papier.

mathematisch model

De flow Q_o door de opening, per breedte-eenheid is:

$$Q_o = C (2gH)^{\frac{1}{2}} \quad (1)$$

met C de effectieve grootte van de opening en H de totale druk boven de opening. (H wordt gemeten in brij-hoogte, d.w.z. de luchtdruk samen met de aanwezige brij geeft een druk, die gelijk is aan de druk, die ontstaan zou zijn wanneer de luchtdruk gelijk is aan de buitendruk en een hoogte H aan brij aanwezig is.)

Verder is:

$$H = L + P - P_o \quad (2)$$

met P en P_o de luchtdruk in de kuip en de buitendruk, gemeten in hoogte brij. L is de werkelijke brijhoogte in de kuip. P_o wordt constant verondersteld.

Voor de lucht in de stortkuip geldt verder:

$$PV = \frac{MRT}{M_w} \quad (3)$$

met V het luchtvolume per breedte-eenheid,

M de luchtmasa per breedte-eenheid,

T de luchttemperatuur (wordt constant verondersteld).

R de gasconstante

en M_w het moleculaire gewicht van lucht.

De tussenwerking van de lucht en de vezelbrij wordt beschreven door:

$$A \frac{dL}{dt} = Q_1 - Q_o \quad (4)$$

$$\frac{dM}{dt} = F_1 - F_o \quad (5)$$

waarbij A het oppervlakte is van de brij per breedte-eenheid,

Q_1 de flow van de brij per breedte-eenheid de kuip in en

Q_o evenzo de kuip uit (dwz. door de opening)

F_1 de flow van de lucht de kuip in en

F_o evenzo de kuip uit.

Linearisatie van deze vergelijkingen geeft (zie /16/):

$$q_o = \frac{Q_o}{2H} h \quad (6)$$

$$h = l + p \quad (7)$$

$$Vp - AP_l = \frac{RT}{M_w} m \quad (8)$$

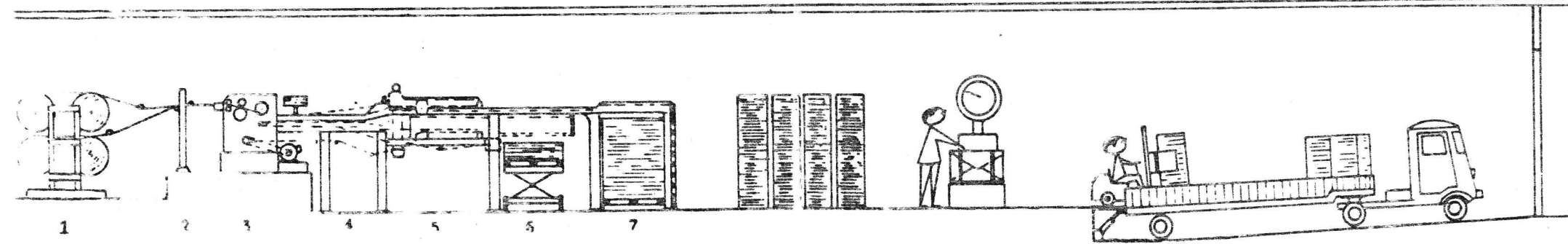
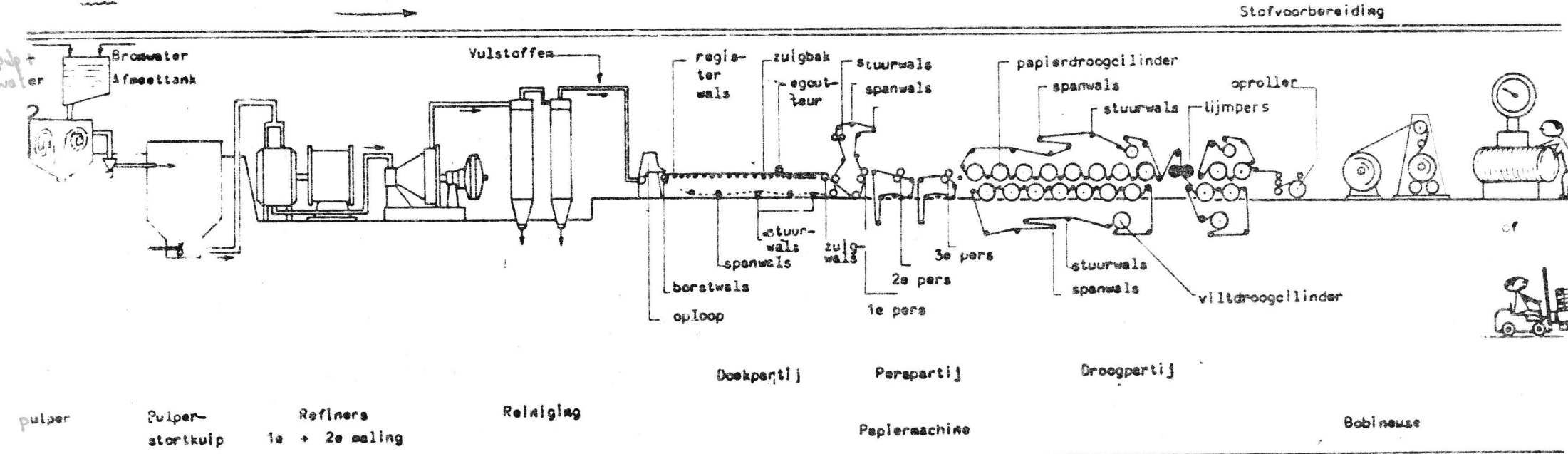
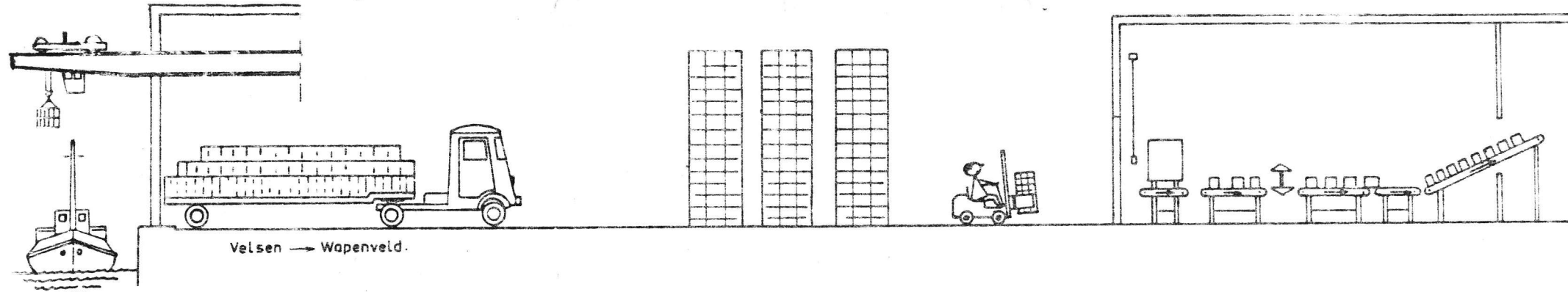
$$A \frac{dl}{dt} = q_1 - q_o \quad (9)$$

$$\frac{dm}{dt} = f_1 - f_o = f_1 - kp \quad (10)$$

waarbij gebruik is gemaakt van $v = -Al$ in (8) en $f_o = kp$ in (10).

Hier zijn kleine letters geschreven voor de overeenkomstige ge-lineariseerde grootheden in hoofdletters.

We gaan ervan uit dat we de instroom van lucht en vezelbrij in de stortkuip kunnen regelen. Voorts beschouwen we q_o, h, l, p en m als toestandsvariabelen. In totaliteit vinden we nu uit de vergelijkingen (6) tem. (10) het volgende lineaire systeem: (Merk op dat we de sturing en toestand dimensieloos hebben gemaakt).



Schematische voorstelling van het fabriceren van papier (VAN GELDER - papier)

papier

Expeditie

$$x = \begin{bmatrix} q_o/Q_o \\ h/H \\ l/L \\ p/P \\ m/M \end{bmatrix} \leftarrow \begin{array}{l} \text{flow van vezelbrij door de opening} \\ \text{totale 'hoogte' van de brij in de kuip} \\ \text{werkelijke hoogte van de brij in de kuip} \\ \text{luchtdruk in de kuip} \\ \text{massa van de lucht} \end{array}$$

$$u = \begin{bmatrix} q_1/Q_o \\ f_1/F_1 \end{bmatrix} \leftarrow \begin{array}{l} \text{instroom van brij in de kuip} \\ \text{instroom van lucht in de kuip} \end{array}$$

$$A = \begin{bmatrix} -\frac{(AP+V)Q_o}{2HAV} & 0 & 0 & -\frac{P^2k}{2HM} & 0 \\ -\frac{(AP+V)Q_o}{HAV} & 0 & 0 & -\frac{P^2k}{HM} & 0 \\ -\frac{Q_o}{AL} & 0 & 0 & 0 & 0 \\ -\frac{Q_o}{V} & 0 & 0 & -\frac{kP}{M} & 0 \\ 0 & 0 & 0 & -\frac{kP}{M} & 0 \end{bmatrix}, B = \begin{bmatrix} \frac{(AP+V)Q_o}{2HAV} & \frac{PF_1}{2HM} \\ \frac{(AP+V)Q_o}{HAV} & \frac{PF_1}{HM} \\ \frac{Q_o}{AL} & 0 \\ \frac{Q_o}{V} & \frac{F_1}{M} \\ 0 & \frac{F_1}{M} \end{bmatrix}$$

$$\text{waarbij } M = \frac{PV}{RT} w.$$

Hierbij stellen hoofdletters de startwaarden voor en kleine letters de verandering. De waarden van A, P, V ect. zijn te vinden in /17/. Substitutie van deze waarden geeft:

$$A = \begin{bmatrix} -0.381 & 0 & 0 & -0.0268 & 0 \\ -0.763 & 0 & 0 & -0.5369 & 0 \\ -0.084 & 0 & 0 & 0 & 0 \\ -0.111 & 0 & 0 & -0.0080 & 0 \\ 0 & 0 & 0 & -0.0080 & 0 \end{bmatrix}, B = \begin{bmatrix} 0.381 & 0.0066 \\ 0.763 & 0.0133 \\ 0.084 & 0 \\ 0.111 & 0.0020 \\ 0 & 0.0020 \end{bmatrix}$$

Omdat de kwaliteit van het papier in hoge mate afhangt van de relatieve snelheid waarmee de brij uit de opening op de band terecht komt, nemen we $z_1 = x_1$ en $z_2 = x_2$ als de te regelen variabelen. Een zo nauwkeurig mogelijke afregeling van de flow door de opening en de druk (lees: snelheid) waarmee de brij uit de opening komt moeten voor een constante kwaliteit papier zorg dragen.

We veronderstellen verder het bestaan van een drietal (onafhankelijke) storings, welke invloed uitoefenen op de toestandsvariabelen x_3 , x_4 en x_5 .

De verstoring d_1 van x_3 is een gevolg van een onnauwkeurigheid in de instroom van brij (u_1). De verstoring d_2 is in eerste instantie

lineair afhankelijk van d_1 , maar bevat nog een tweede component die het gevolg is van fluctuaties in de (constant veronderstelde) buitendruk P_o . De verstoring d_3 tenslotte, is behalve afhankelijk van d_2 ook een gevolg van onnauwkeurigheden in de instroom van lucht in de kuip (u_2). Door eventueel een basistransformatie uit te voeren op de storings-ruimte \mathbb{R}^3 zijn d_1 , d_2 en d_3 als onafhankelijk te beschouwen.

Als waarnemingen kiezen we in eerste instantie x_1 en x_3 .

Op deze manier verkrijgen we de volgende matrices:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}, G = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

LIDIA berekent nu de volgende deelruimten:

$$V^{\#} \ker H = \text{sp}(e_3, e_4, e_5)$$

$$R^{\#} \ker H = 0$$

$$S^{\#} \text{im } G = \text{sp}(g_1, e_3, e_4, e_5)$$

$$N^{\#} \text{im } G = \mathbb{R}^5, \text{ met } g_1 = \text{sp} \left(\begin{bmatrix} 0.0499 \\ 0.9988 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right)$$

en concludeert oplosbaarheid van DDP. De bijbehorende oplossing is:

$$F = \begin{bmatrix} 0 & 0 & 0 & -126.172 & 0 \\ 0 & 0 & 0 & 7261.799 & 0 \end{bmatrix}$$

DDEP en DDPM blijken niet oplosbaar.

Opnieuw proberen we door het toevoegen van een extra waarneming of DDPM dan oplosbaar is.

In eerste instantie proberen we $y_3 = x_2$. LIDIA berekent dezelfde deelruimten, zodat deze y_3 geen beter resultaat oplevert.

Hetzelfde is het geval met $y_3 = x_5$.

De extra waarneming $y_3 = x_4$ geeft extra wel het beoogde resultaat.

LIDIA berekent nu:

$$V^{\#} \ker H = \text{sp}(e_3, e_4, e_5)$$

$$R^{\#} \ker H = 0$$

$$S^{\#} \text{im } G = \text{sp}(e_3, e_4, e_5)$$

$$N^{\#} \text{im } G = \mathbb{R}^5.$$

Zowel DDEP als DDPM blijken nu oplosbaar te zijn.

Helaas is (A,B) niet regelbaar en (A,C) niet waarneembaar zodat de equivalente problemen met poolplaatsing nimmer oplosbaar zijn.

conclusies

Het is mogelijk een constante kwaliteit papier te verkrijgen door de storingen die l , p en m beïnvloeden te ontkoppelen. Dit is zelfs mogelijk ^{door} slechts q_0 , l en p waar te nemen. (DDPM). Op grond van dezelfde waarnemingen is het mogelijk een schatting te maken van q_0 en h , die beide onafhankelijk zijn van de storingen in l , p en m . (BDEP). Rosenbrock heeft in /17/ een andere methode toegepast. Zijn methode ging niet uit van storingen. Een vergelijking tussen beide methoden voert in deze context te ver. Bovendien gaan beide methoden uit van geheel verschillende startpunten.

LIJST VAN GEBRUIKTE SYMBOLEN

$\text{sp}(e_1, \dots, e_k)$	$= \{x \mid x = \lambda_1 e_1 + \lambda_2 e_2 + \dots + \lambda_k e_k\}$
$\text{im } A$	$= \{x \mid \exists y : x = A y\}$
$\ker A$	$= \{y \mid Ay = 0\}$
$\text{spec}(A)$	$= \{\lambda \mid A - \lambda I = 0\}$ (polen van A)
$\chi_A(\lambda)$	$= A - \lambda I $ (karakteristiek polynoom van A)
$\langle A \mid B \rangle$	$= \sum_{i=1}^{n-1} A^i B$
$\langle K \mid A \rangle$	$= \bigcap_{i=1}^{n-1} A^{-i} K$
$\mathcal{V}(K)$	$= \{V \mid A \cdot V \subset V + \text{im } B \text{ en } V \subset K\}$
$\mathcal{R}(K)$	$= \{R \in \mathcal{V}(K) \mid \text{spec}((A + B F) R) \text{ specificeerbaar}\}$
$\mathcal{S}(K)$	$= \{S \mid A \cdot (S \cap \ker C) \subset S \text{ en } K \subset S\}$
$\mathcal{N}(K)$	$= \{N \in \mathcal{S}(K) \mid \text{spec}((A + L C)(\text{mod } N)) \text{ specificeerbaar}\}$
$\underline{V}^*(K)$	$= \sup \{V \in \mathcal{V}(K)\}$
$\underline{R}^*(K)$	$= \sup \{R \in \mathcal{R}(K)\}$
$\underline{S}^*(K)$	$= \inf \{S \in \mathcal{S}(K)\}$
$\underline{N}^*(K)$	$= \inf \{N \in \mathcal{N}(K)\}$
$\underline{F}(V)$	$= \{F \mid (A + B \cdot F) \cdot V \subset V\}$
$\underline{L}(S)$	$= \{L \mid (A + L \cdot C) \cdot S \subset S\}$
$\underline{K}(V, S)$	$= \{K \mid (A + B \cdot K \cdot C) \cdot S \subset V\}$
$\underline{F}_p(R)$	$= \{F \in \underline{F}(R) \mid \text{spec}((A + B F) R) \text{ specificeerbaar}\}$
$\underline{L}_p(N)$	$= \{L \in \underline{L}(N) \mid \text{spec}((A + L C)(\text{mod } N)) \text{ specificeerbaar}\}$

LIJST VAN GEBRUIKTE AFKORTINGEN

ACSA	- Allmost Constrollability Subspace Algorithm (15,26)
ADDEP	- Allmost Disturbance Decoupling Estimation Problem (4)
ADDP	- Allmost Disturbance Decoupling Problem (4)
ADDFM	- Allmost Disturbance Decoupling Problem with Measurements (4)
CSA	- Controllability Subspace Algorithm (52)
DDEP	- Disturbance Decoupling Estimation Problem (4,44)
DDEPPP	- Disturbance Decoupling Estimation Problem with Pole-Placement (44)
DDP	- Disturbance Decoupling Problem (4,42)
DDPM	- Disturbance Decoupling Problem with Measurements (4,46)
DDPMPP	- Disturbance Decoupling Problem with Measurements and Pole Placement (46)
DDPPP	- Disturbance Decoupling Problem with Pole Placement (42)
ISA	- Invariant Subspace Algorithm (15,21)
LIDIA	- LInear systems DIsturbance decoupling Algorithms (3)
LINPAK	- LInear systemen computerPAKket (3)
SVD	- Singular Value Decomposition (19)

De getallen tussen haakjes geven het bladzijnummer aan, waar een definitie of uitleg te vinden is.

REFERENTIES

- /1/ G. Kamphuis.
Ontwerp van een interactief computerpakket voor het ontwerpen en analyseren van lineaire stationaire regelsystemen. afstudeerverslag, okt. 1979.
- /2/ W.M. Wonham.
Linear multivariable control, a geometric approach.
Springer-Verlag 1979
- /3/ P.M. van Dooren.
The generalized eigenstructure problem in linear system theory.
IEEE transactions on automatic control, vol. AC 26, no. 1, 1981, pp 111-129.
- /4/ J.C. Willems & C. Commault.
Disturbance Decoupling by measurement feedback with stability or pole placement.
Siam J. Control and optimisation, vol. 19, no. 4, juli 1981. pp 490-504.
- /5/ J.M. Schumacher.
Compensator synthesis using (C,A,B) pairs.
IEEE transactions on automatic control, vol. AC 25, no. 6, dec. 1980, pp 1133-1138.
- /6/ J.M. Schumacher.
Dynamic feedback in finite- and infinite dimensional linear systems.
proefschrift, Mathematisch Centrum Amsterdam 1981.
- /7/ J.C. Willems
Almost noninteracting control design using dynamic state feedback.
Analysis and optimisation of systems, lecture notes in control and information sciences, vol. 28, 1980, Springer-Verlag, pp. 555-561.
- /8/ C.H. Lindsey & S.G. van der Meulen.
Informal introduction to Algol 68 (revised).
North Holland 1980.
- /9/ S.G. van der Meulen & M. Veldhorst.
Torrix, vol. 1.
Mathematical Centre Tracts 86, Mathematisch Centrum Amsterdam
- /10/ Cyber loader version 1, reference manual
- /11/ J.C. Willems.
systeemtheorie.
collegedictaat. RuG Groningen.

- /12/ J.C. Willems
Almost invariant subspaces: an approach to high gain feedback design - part 1 : almost controlled invariant subspaces.
IEEE transactions on automatic control, vol. AC 26, no. 1, 1981, pp 225-252.
- /13/ J.C. Willems
Almost invariant subspaces: an approach to high gain feedback design - part 2: almost conditional invariant subspaces.
binnenkort in: IEEE transactions on automatic control.
- /14/ Takamatsu, Hashimoto, Nakai.
A geometric approach to multivariable control design of a distillation column.
IFAC seventh triennial world congress, vol. 1, pp 309-318.
Pergamon press 1978.
- /15/ Aström, Källström.
Identification of ship steering dynamics.
Automatica vol. 12, 1976, pp. 9 e.v.
- /16/ J.K. Zuidweg.
Automatic guidance of ships as a control problem.
proefschrift 1970, Waltman Delft.
- /17/ H.H. Rosenbrock
Computer-aided control design.
Academic press 1974.
- /18/ Källström, Aström, Thorell, Eriksson, Sten.
Adaptive Autopilots for tankers.
Automatica, vol. 15, 1979, pp. 241-254.

appendix A: PROGRAMMEREN IN ALGOL 68

De hiervoor beschreven implementatie is op de CYBER 170/760 van het rekencentrum van de RUG uitgevoerd in de programmeertaal ALGOL 68. De keuze van juist deze programmeertaal stoelt op verschillende gronden:

- aanvulling in dezelfde taal
- groot scala van mogelijkheden
- gebruik van TORRIX

Aangezien Geert Kamphuis (zie /1/) het pakket LINPAK reeds in ALGOL 68 heeft geschreven en deze implementatie bedoeld is als uitbreiding van LINPAK is het gebruik van dezelfde programmeertaal een voor de hand liggende keuze (zie echter noot).

ALGOL 68 komt, onder de op dit moment beschikbare talen, het meest in de richting van de wiskundige schrijfwijze. Bijna alle wiskundige formuleringen (zeker die in de lineaire algebra en de systeemtheorie) kunnen betrekkelijk eenvoudig worden "vertaald" in ALGOL 68 - termen. Dit vindt zijn reden in het feit dat in ALGOL 68 zelf operatoren te definiëren zijn.

Door de keuze van ALGOL 68 is het gebruik van TORRIX (vector-matrix-rekening, zie /9/) mogelijk waardoor zonder extra programmeren zeer doeltreffend met matrices en vectoren is te werken. Bovendien bestaat er een goede interface met FORTRAN (zeker beter dan de beperkte interface PASCAL-FORTRAN) waardoor eenvoudig FORTRAN-routines (uit bijvoorbeeld de NAG-bibliotheek) zijn aan te sluiten.

Eenmaal werkend in ALGOL 68 komen echter twee belangrijke nadelen c.q. problemen naar voren:

- programma's worden snel te groot om in z'n geheel geladen te kunnen worden
- de ALGOL 68 - compiler is niet fout-vrij.

geheugengebruik

Al betrekkelijk snel wordt een ALGOL 68 programma te groot om in een keer in het geheugen te kunnen worden geladen. Vooral indien interactief wordt gewerkt, wordt de dan geldende limiet van 120.000 words al met een relatief klein programma bereikt.

Overigens is dit niet in de eerste plaats een tekortkoming van de programmeertaal, doch veel eerder van de gebruikte computer. Zo kent de DEC-10 (TH-Twente) deze tekortkoming niet, omdat deze machine beschikt over zg. 'virtueel' geheugen. Zonder consequenties voor de gebruiker (alles wordt intern geregeld) kunnen programma's (zeer) groot worden.

Een ALGOL 68 programma genereert in de regel echter wel meer code dan een vergelijkbaar programma in bijv. PASCAL. Dit is een gevolg van het feit, dat de mogelijkheden van ALGOL 68 nu eenmaal veel groter zijn ("makkelijker programmeren betekent meer werk en soms wat meer code voor de vertaler"). Een PASCAL-programma, met vergelijkbare bedoelingen als het huidige LIBIA, zal echter ook zeker niet zonder meer kunnen draaien.

De oplossing voor dit probleem is het gebruik van de segment-loader op de CYBER. Een en ander betekent dan wel extra voorzichtig programmeren en op zich staande procedures zo klein mogelijk houden, opdat aparte segmenten niet te groot worden. Hoe een en ander in detail is opgelost (o.a. met het inlezen van commando's en het starten van het juiste programma-onderdeel) wordt uitvoerig beschreven in /1/.

compiler-fouten

Omdat ALGOL 68 weinig gebruikers kent (hoewel een vooruitstrevende taal is ALGOL 68 waarschijnlijk te laat op de markt verschenen en vormen andere talen, o.a. ADA, een te grote concurrent) komen fouten in de vertaler slechts moeizaam aan het licht. Bovendien komt men in Den Haag bij CDC niet snel met verbeteringen aan de huidige vertaler op de CYBER. De laatste twee jaar zijn door het rekencentrum een tiental fouten doorgegeven, waarvan er tot nu toe geen enkele is verbeterd.

Bovenstaande punten zijn er de oorzaak van dat erg veel tijd gestoken moet worden in het uittesten van ALGOL 68 programma's.

vergelijking.

Er is voor ALGOL 68 veel tijd nodig om een goed inzicht in de taal te verkrijgen, ondanks het feit, dat de taal nagenoeg orthogonaal is opgezet ("alles mag"). Een taal als PASCAL is, ondanks de vele uitzonderingen veel sneller te leren. Hier staat tegenover dat PASCAL veel beperkter is en indien het inzicht in ALGOL 68 er eenmaal is, programmeert men in ALGOL 68 veel gemakkelijker dan in PASCAL.

Vanwege de uitgebreidheid van de taal kost het uittesten van een ALGOL 68 programma wel veel tijd. Soms zijn kleine typefouten (= ipv. :=, een haakje te veel of te weinig) oorzaak van (veel), soms op heel andere plaatsen optredende en soms heel vreemde foutmeldingen.

Bovendien speelt in dit opzicht het feit, dat de vertaler nog niet foutvrij is, in sterke mate mee. (Ter verduidelijking: het ontdekken van een fout in de vertaler, alsmede het vinden van een oplossing om 'om de fout heen te programmeren' kostte mij eens drie weken werk).

conclusies.

Programmeren in ALGOL 68 heeft als voordeel dat wiskundige formuleringen vrij eenvoudig in de taal zijn om te zetten. Samen met TORRIX en de goede interface met FORTRAN kan dan vrij eenvoudig en snel worden geprogrammeerd. Bovendien is zo'n programma, mits niet al te ingewikkelde constructies worden gebruikt, vrij goed leesbaar (in ALGOL 68 kan echter ook onleesbaar worden geprogrammeerd door het gebruik van zeer ingewikkelde en compacte constructies).

Na de programmeerfase treden echter de nadelen sterk na voren. Het gebruik van de segment-loader is (althans op de CYBER) vrijwel onvermijdelijk en het opsporen van fouten kan erg veel tijd kosten.

Het is daarom aan mensen met weinig programmeerervaring aan te raden vooral deze nadelen in ogenschouw te nemen en is het het overwegen waard de mindere uitgebreidheid van PASCAL te verkiezen boven de veelzijdigheid van ALGOL 68.

Indien men echter een behoorlijke programmeerervaring heeft, is, m.i. ALGOL 68 een bijzonder mooie taal, waarin de ingewikkelste problemen zich vrij eenvoudig laten programmeren.

Helaas is er een grote onzekerheid over de ontwikkeling van ALGOL 68 in de toekomst en kan deze onzekerheid een reden zijn om toch maar bijv. PASCAL te verkiezen, daar de toekomst van deze taal in ieder geval gewaarborgd is.

noot.

Het is echter niet persé noodzakelijk dezelfde programmeertaal te gebruiken. In deze context past het idee van H.C.Croon.

In plaats van één groot pakket (zoals nu LINPAK) wordt uitgegaan van een hoofdprogramma, dat commando's kan inlezen van het scherm en aan de hand daarvan een CCL-procedure schrijft. M.b.v. zo'n procedure kunnen dan bepaalde programma's worden gestart. Datagegevens kunnen m.b.v. text-files van het ene naar het andere programma worden overgedragen. Deze werkwijze heeft het voordeel dat elk (sub-)programma geschreven kan worden in een voor dat doel meest geschikte of gewenste programmeertaal.

appendix B. IMPLEMENTATIE EN HANDLEIDING

implementatie

Voor de implementatie van LIDIA is gekozen voor de programmeertaal Algol68. Zie appendix A.

De belangrijkste moeilijkheid, die zelfs de uitvoering van LIDIA zou kunnen belemmeren, is het operating system van de CYBER. Voordat een programma op de CYBER kan starten moet deze in zijn geheel zijn geladen. Het geheugengebruik voor terminal-gebruikers (interactief) is 120000 woorden en deze limiet wordt al na implementatie van een deel van LIDIA ruim overschreden.

Een oplossing voor dit probleem werd gevonden door gebruik te maken van de segmentloader, samen met een aangepaste structuur van LIDIA. Het programma werd opgesplitst in onderdelen die wel en die niet gelijktijdig in het geheugen behoeven te zijn en aan de hand van een opgestelde boom van procedures zorgt de segmentloader er voor dat aangeroepen procedures in het geheugen worden geladen en andere daar weer uit worden verwijderd.

Een probleem hierbij is dat een tak van de boom te groot kan worden. Hiervoor is de constructie aangepast als volgt:

Procedures die elkaar aanroepen doen dat niet direct, doch via globale variabelen, welke daarvoor in de eigen prelude van LIDIA moeten zijn gedeclareerd. Deze globale variabelen zijn als pointers uitgevoerd ('ref' 'ref') om onnodig lang de stack bezet te houden te voorkomen (de garbage collector ruimt ze nu op als de verbinding tussen pointer en waarde wordt verbroken).

beschrijving.

Elke module van het pakket bestaat uit een aantal commando's, een prelude en een postlude. Elk van deze commando's kan op zich weer een andere module aanroepen of een rechtstreekse actie tot gevolg hebben. Bij binnenkomst van een module wordt eerst de bij die module behorende prelude aangeroepen en deze test eventueel of aan de benodigde voorwaarden is voldaan om de module werkelijk binnen te gaan.

Als een module wordt verlaten wordt de bij die module behorende postlude aangeroepen. Deze kan, indien nodig, globale variabelen op 'NIL' zetten.

Het aanroepen van een of meer procedures, waaruit een rechtstreekse actie bestaat, vindt plaats in de procedure CHPROC.

Aan de hand van de globale variabele PROCNR, die in de LIDIA-prelude is opgenomen, kiest CHPROC mbv. een CASE-statement de bij PROCNR behorende actie, maar geeft PROCNR eerst de waarde 0. Tijdens zo'n actie kan PROCNR opnieuw een waarde krijgen. Hierdoor bestaat dan de gewenste mogelijkheid dat procedures elkaar indirect aanroepen.

Als na een actie PROCNR = 0, dan wordt CHPROC verlaten en kan in de procedure CSI een nieuw commando worden ingelezen.

Alle proc's die in het CASE-statement van CHPROC worden aangeroepen zijn in de LIDIA-prelude als externe procedures gedeclareerd.

```
'PROC' CHPROC = 'VOID' ;
(
  'INT' P := PROCNR ; PROCNR := 0 ;
  'WHILE' P /= 0 'DO'
    'CASE' P 'IN'
      ...
      ...
      ...
    'ESAC' ;
  P := PROCNR ; PROCNR := 0 ;
  'OD'
) ;
```

De commando's worden ingelezen door de procedure CSI (Command String Interpreter). Deze vergelijkt het gelezen commando met de commando's die in de betreffende module mogelijk zijn. Als een overeenkomstige commando wordt gevonden, wordt of een actie ondernomen, dwz. PROCNR krijgt een nieuwe waarde en CHPROC wordt aangeroepen, of een nieuwe module wordt aangeroepen. CSI moet als inputparameter de bij de module behorende data hebben. Daarom zijn deze gegevens per module samengevoegd in een variabele:

```
[1:n] 'COMMAND' MODULE ;
waarbij de 'MODE' 'COMMAND' is gedeclareerd als:
'MODE' 'COMMAND' = ('STRING' S,
                    'UNION' ('REF' [] 'COMMAND',
                              'INT') NEXT ) ;
```

waarbij S het commando is voor een volgende module of rechtstreekse actie.

Hierbij is NEXT 'OF' MODULE [1] altijd een integer, die de prelude voor MODULE aangeeft en NEXT 'OF' MODULE [2] een integer voor de postlude van MODULE. De overeenkomstige strings zijn ":", resp. "HELP", waardoor deze twee bij het commando HELP ook automatisch worden afgedrukt. Daar apart op deze twee wordt getest, kunnen de commando's ":" en HELP nooit de prelude of postlude direct aanroepen. Zie voor een gedetailleerde beschrijving de listing van CSI. De verschillende 'REF' [] 'COMMAND' s zijn gedeclareerd en geïnstalleerd in de initialisatieprocedure INIT LIDIA, waarin tevens de stringterminators worden gedeclareerd. De globale variabelen en modes zijn opgenomen in de LIDIA-prelude.

overzicht.

Het totale LIDIA pakket is 3153 regels groot. De grootst mogelijke geheugenruimte is dankzij het gebruik van de segmentloader (en het gebruik van LEVELs daarin) teruggebracht van veel te groot tot 71732 (oct) woorden. Het resterende open geheugen was nu groot genoeg om alle gegevens (globale variabelen) nodig om analyse te plegen op de destillatiekolom van blz. 58 op te slaan. De berekening van de resultaten uit die paragraaf alsmede het afdrukken daarvan nam in totaal ca. 7 cp-seconden in, hetgeen ruim voldoende was om onder de limiet van 16 cp.seconden te blijven.

De file met de absolute versie van LIDIA tenslotte is ca. 850 PRU's groot. Hierdoor kan deze file niet meer als IA-file (de zg. GET-files) worden weggezet, maar uitsluitend nog als DA-file (ATTACH-file).

handleiding.

De handleiding van LIDIA is apart bijgevoegd.